H. IONESCU • C. DINESCU • V. BURLACU

teoria grafelor

cu unele aplicații în economie

EDITURA ȘTIINȚIFICĂ

H. IONESCU • C. DINESCU • V. BURLACU

teoria grafelor

cu unele aplicații în economie

EDITURA ȘTIINȚIFICĂ
București • 1969

Scanned with OKEN Scanner

PREFAȚĂ

O caracteristică distinctă a revoluției tehnico-științifice din epoca modernă pe care o trăim o constituie folosirea științei în aproape toate domeniile de activitate. Organizarea științifică a producției și muncii, fundamentarea teoretică a conducerii și planificării economiei naționale, introducerea pe scară din ce în ce mai largă a progresului tehnic și științific, capătă în momentul de față o extindere și o amploare nemaiîntîlnite în țara noastră.

Complexitatea proceselor economice, cerințele mereu crescînde ale conducerii și planificării științifice a economiei naționale într-un avînt fără precedent, impun cunoașterea soluțiilor optime din punct de vedere economic, cu maximum de precizie. În soluționarea unor astfel de probleme complexe nu mai pot fi utilizate metodele empirice care pot conduce la pagube materiale considerabile, ci trebuie folosite metode matematice moderne și mașini electronice de calcul capabile să execute în timp util un număr cît mai mare de operații într-un timp cît mai scurt. Se poate firma că în momentul de față acțiunea de folosire a matematicilor moderne se extinde cu maximum de eficiență și în domeniul economic, la toate nivelele de decizie, punînd în circuitul cercetării științifice discipline noi cum ar fi de pildă cibernetica economică, cercetarea operațională sau programarea calculatoarelor electronice. Se dovedește în mod practic faptul că în procesul de producție se obțin rezultate superioare prin folosirea mijloacelor tehnice moderne de calcul și automatizare, precum și a instrumentelor de analiză pe care le oferă cele mai noi domenii ale matematicii, ale cibernetici economice și informaticii economice.

Scanned with OKEN Scanner

În acest vast program de acțiune privind introducerea celor mai noi cuceriri ale științei mondiale în economie, se înscriu o serie de acțiuni însemnate inițiate de Academia Republicii Socialiste România, Consiliul Național al Cercetării Științifice, institute de învățămînt superior, universități, institute de cercetări departamentale ș.a. În revistele de specialitate apar tot mai bogate materiale tratînd probleme noi de aplicare a matematicilor moderne în economie. Congrese naționale sau internaționale, colocvii, simpozioane pe aceste teme noi, au intrat deja în preocupările de bază ale matematicienilor români care își aduc aportul lor în această direcție.

În acest context, autorii lucrării de față s-au străduit să elaboreze un material privind aplicațiile teoriei grafelor în unele probleme economice.

Materialul, după părerea autorilor, poate să fie util specialiștilor care lucrează în diverse domenii de activitate economică, precum și studenților din facultățile economice.

■

Grafele se întîlnesc în cele mai diverse domenii de activitate, care pot fi concretizate prin scheme orientate, cum sînt probleme de repartiție, probleme de planificare, conducere și control, probleme de psihologie și altele tot atît de importante.

Varietatea situațiilor în care apare graful a impus abstractizarea acestei noțiuni, ajungîndu-se în cele din urmă la o disciplină de sine stătătoare.

Așa se explică faptul că astăzi grafele sînt folosite cu succes în discipline matematice ca algebra, calculul probabilităților, teoria funcțiilor etc.

Metodele folosite de teoria grafelor permit o soluționare intuitivă și, ca atare, ușoară și rapidă, a unor probleme complexe care, altfel, ar necesita operații multiple și abstracte.

Se poate spune, pe drept cuvînt, că teoria grafelor poate fi folosită totdeauna în probleme în care se cere optimizarea unei acțiuni.

Ca disciplină de sine stătătoare, teoria grafelor s-a constituit destul de recent și cu toate acestea, datorită vastului

Scanned with OKEN Scanner

cîmp de aplicabilitate, precum și datorită procedeelor eficace de investiție, ce ajută intuiția, ea a luat o mare amploare, constituind la ora actuală un instrument de lucru util pentru economiști, ingineri, tehnicieni etc.

Menționăm că pentru a veni în ajutorul acelora care nu au la îndemînă întregul aparat matematic necesar înțelegerii materialului expus, am redat aceste noțiuni matematice fie acolo unde intervin direct, fie în anexa lucrării.

Vom fi recunoscători tuturor celor care ne vor semnala eventualele lacune ale lucrării și ne vor ajuta pentru îmbunătățirea ei.

AUTORII

Scanned with OKEN Scanner

CAPITOLUL

1

NOȚIUNI DE BAZĂ ÎN TEORIA GRAFELOR

§ 1. *Introducere*

Teoria grafelor este o ramură a teoriei mulțimilor, destul de recentă, care s-a dovedit foarte utilă și cu aplicații în domenii variate: economie, chimie organică, organizare, psihologie, anumite domenii ale artei etc. În scopul familiarizării cititorilor cu diversele domenii de aplicații, vom prezenta cîteva probleme care pot fi studiate și rezolvate cu ajutorul teoriei grafelor.

Exemplul 1. Să presupunem că se pune problema construirii unei șosele între două centre industriale, notate cu x_0 și x_{11}, care ar putea să treacă prin localitățile x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}. Cunoscînd costul lucrării pentru porțiunile de șosea care leagă între ele diverse localități (x_i), să se determine traseul șoselei dintre localitățile x_0 și x_{11} astfel încît cheltuielile totale pentru efectuarea lucrării să fie minime.

Dacă reprezentăm localitățile prin puncte, iar porțiunile de șosea prin arce, atunci obținem schema din figura I.1.1. Putem presupune că ar fi vorba de o regiune muntoasă cu șosele înguste pe care sensul de circulație este unic, de unde rezultă orientarea în sens unic a arcelor de pe figură.

Exemplul 2. O secție a unei întreprinderi are de executat cinci produse diferite. Produsele respective nu pot fi executate simultan, iar trecerea de la executarea unui produs la altul implică anumite cheltuieli. Se pune problema determinării unei ordini de lansare în fabricație, într-o asemenea manieră, încît cheltuielile totale de trecere să fie minime. De această dată, dacă vom reprezenta prin puncte produ-

Scanned with OKEN Scanner

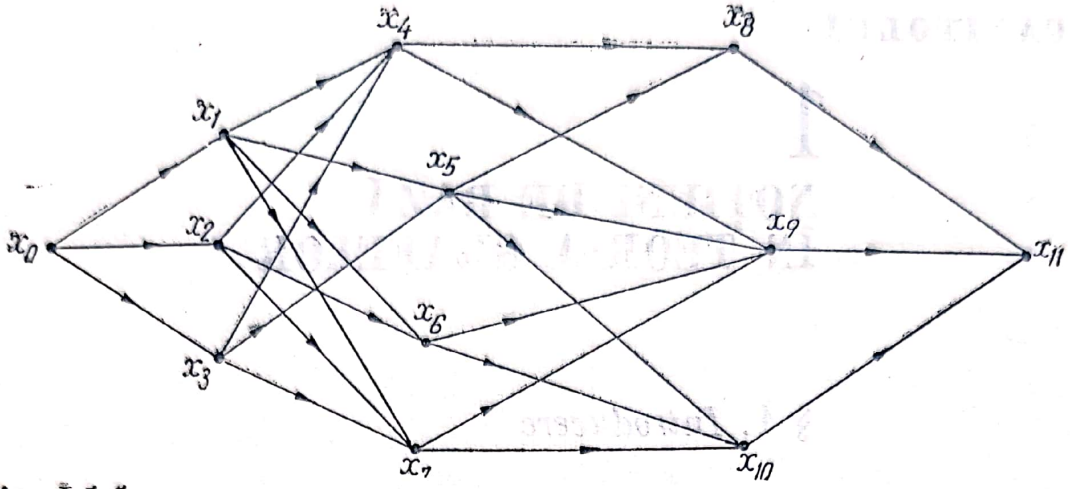


Fig. I.1.1

sele, iar prin arce posibilitățile de trecere de la fabricarea unui produs la altul, se obține schema din figura I.1.2.

Exemplul 3. Fie dată pentru executare o lucrare care necesită 12 operații intermediare. Pe baza observațiilor statistice anterioare, se cunosc duratele operațiilor ce compun lucrarea, precum și condiționările dintre ele în sensul că unele nu pot începe pînă ce altele nu au fost executate.

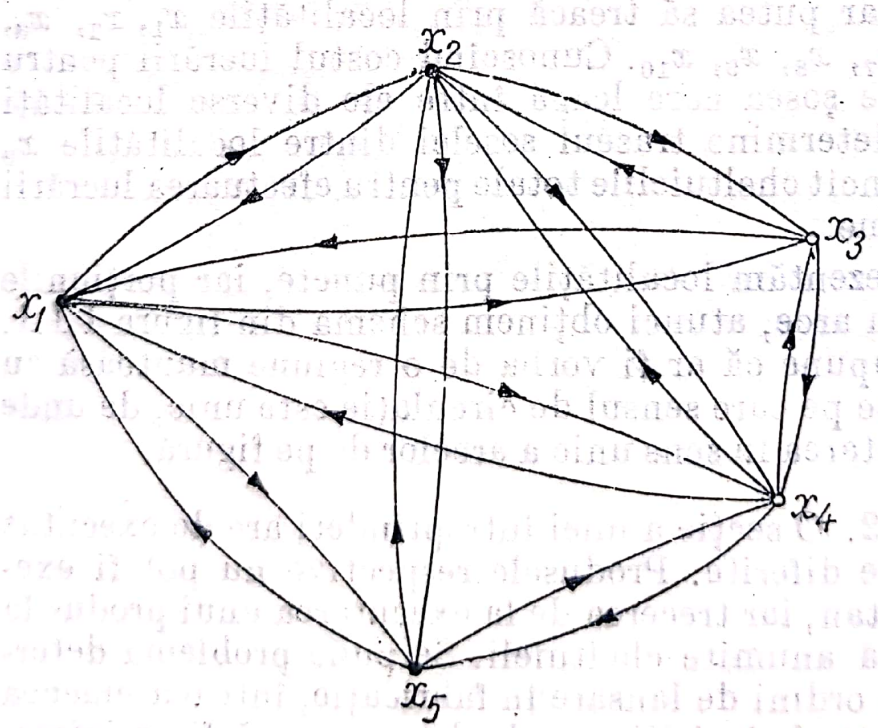


Fig. I.1.2

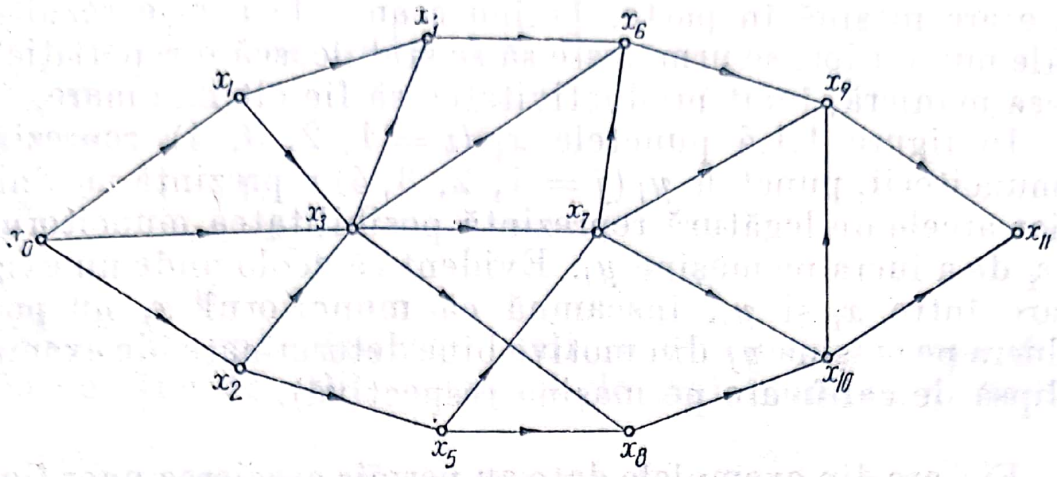
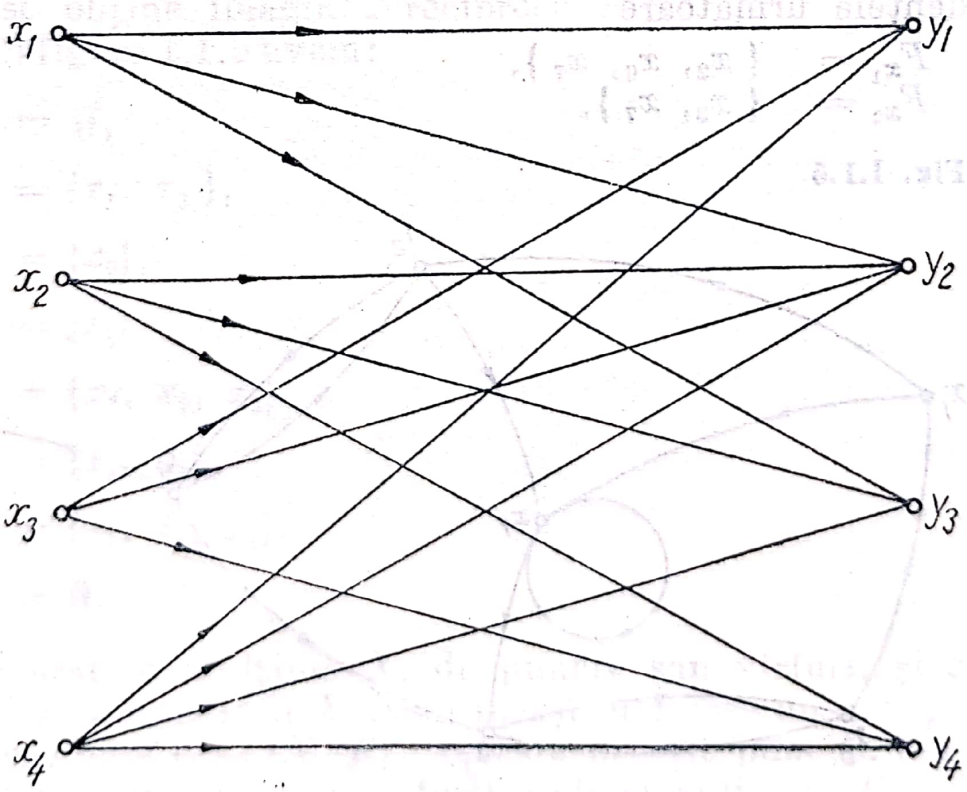


Fig. I.1.3

Problema permite atașarea figurii I.1.3, în care reprezentăm prin puncte operațiile și prin arce orientate condiționările respective.

Exemplul 4. Să admitem că patru muncitori urmează să fie repartizați să lucreze pe patru mașini. Din practică se cunoaște randamentul pe care-l dă fiecare muncitor, pentru

Fig. I.1.4



fiecare mașină în parte. Ținînd seama de aceste rezultate ale muncii lor, se urmărește să se stabilească o repartiție de așa manieră, încît productivitatea să fie cît mai mare.

În figura I.1.4 punctele x_i $(i = 1, 2, 3, 4)$ reprezintă muncitorii, punctele y_j $(j = 1, 2, 3, 4)$ reprezintă mașinile, iar arcele de legătură reprezintă posibilitatea muncitorului x_i de a lucra pe mașina y_j. Evident că acolo unde nu există arc între x_i și y_j, înseamnă că muncitorul x_i nu poate lucra pe mașina y_j din motive bine determinate (de exemplu lipsă de calificare pe mașina respectivă).

Fiecare din exemplele date au permis asocierea unor figuri constituite din puncte și arce orientate (săgeți) care unesc perechi din aceste puncte.

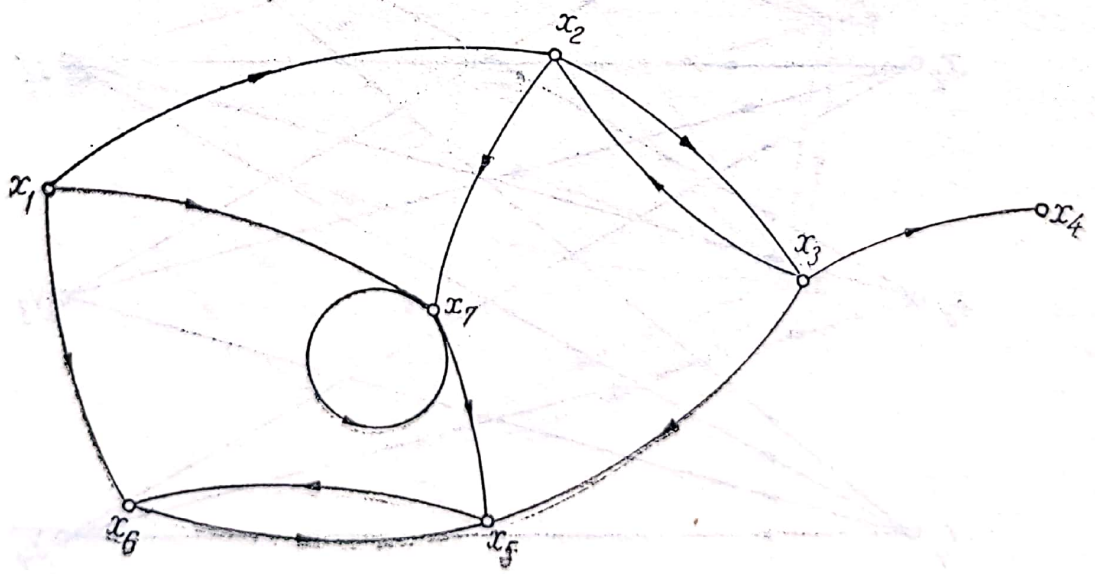
În sens matematic, aceste săgeți pot simboliza o aplicație F pentru care extremitatea inițială a arcului reprezintă argumentul sau variabila, iar extremitatea finală a aceluiași arc reprezintă imaginea sau valoarea aplicației.

Tot din figuri reiese că oricărui punct (sau vîrf) al mulțimii $X = \{ x_i \}$ îi corespunde, prin aplicația F, o submulțime a aceleiași mulțimi X.

Astfel, dacă considerăm figura I.1.5, distingem corespondențele următoare

$$F_{x_1} = \{ x_2, x_6, x_7 \},$$
$$F_{x_2} = \{ x_3, x_7 \},$$

Fig. I.1.5



$F_{x_3} = \{x_2, x_4, x_5\}$,
$F_{x_4} = \emptyset$,
$F_{x_5} = \{x_6\}$,
$F_{x_6} = \{x_5\}$,
$F_{x_7} = \{x_5, x_7\}$,
$F_{x_8} = \emptyset$.

Uneori, în aplicații, este util să se cunoască semnificația notației $F^k_{x_i}$. Din exemplul de mai înainte s-a văzut că

$$F_{x_7} = \{x_5, x_7\}.$$

Prin $F^2_{x_7}$ vom înțelege

$$F^2_{x_7} = F(F_{x_7}) = F_{x_5} \cup F_{x_7} = \{x_6\} \cup \{x_5, x_7\} = \{x_5, x_6, x_7\}.$$

Analog,

$$F^3_{x_7} = F(F^2_{x_7}) = F_{x_5} \cup F_{x_6} \cup F_{x_7} = \{x_6\} \cup \{x_5\} \cup \{x_2, x_7\} = \{x_5, x_6, x_7\}.$$

În general

$$F^k_{x_i} = F\left(F^{k-1}_{x_i}\right).$$

Dacă în figura I.1.5 se schimbă sensul tuturor săgeților, atunci se obține imaginea reciprocă F^{-1}.

Pentru figura I.1.5 avem:

$F^{-1}_{x_1} = \emptyset$,
$F^{-1}_{x_2} = \{x_1, x_3\}$,
$F^{-1}_{x_3} = \{x_2\}$,
$F^{-1}_{x_4} = \{x_3\}$,
$F^{-1}_{x_5} = \{x_6, x_7, x_3\}$
$F^{-1}_{x_6} = \{x_1, x_5\}$,
$F^{-1}_{x_7} = \{x_1, x_2, x_7\}$,
$F^{-1}_{x_8} = \emptyset$.

În general, o mulțime X, de puncte sau vîrfuri, și o aplicație F, a mulțimii X în ea însăși, definesc un graf pe care-l vom nota $G = (X, F)$. Această notație pune în evidență atît mulțimea X a vîrfurilor cît și aplicația F.

Scanned with OKEN Scanner

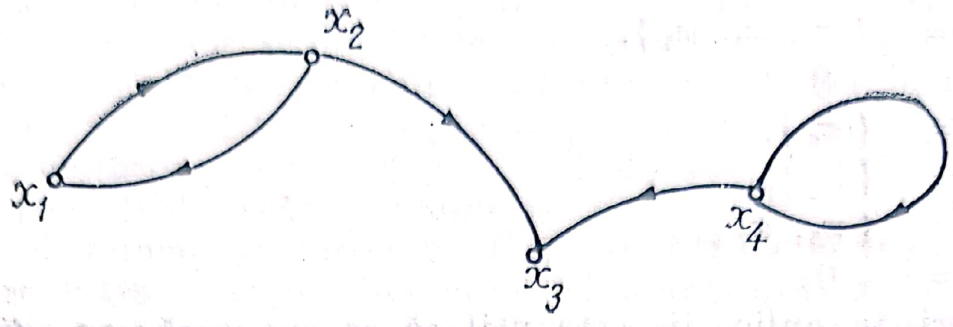


Fig. I.1.6

Deoarece aplicația F este reprezentată simbolic prin arce orientate, graful G mai poate fi pus în evidență de cuplul format din mulțimea X a vîrfurilor și mulțimea A a arcelor, adică $G = (X, A)$.

Dacă într-un graf numărul vîrfurilor sale, notat prin $|X|$, este finit, graful se numește *graf finit*.

Graful definit de mulțimea X a vîrfurilor și de mulțimea A a arcelor sale se numește *graf orientat*.

Între două vîrfuri ale unui graf există o muchie, dacă există cel puțin un arc care să le lege într-un sens sau altul.

Graful definit de mulțimea X a vîrfurilor sale și de mulțimea muchiilor sale se numește *graf neorientat*. De exemplu, în figura I.1.6 deși există cinci arce, nu avem decît trei muchii, deoarece arcele (x_1, x_2) și (x_2, x_1) conduc la o singură muchie, iar bucla[1] (x_4, x_4), nu este o muchie.

Oricare ar fi graful asociat unei probleme concrete, se poate defini noțiunea de *graf parțial* al grafului dat, anume,

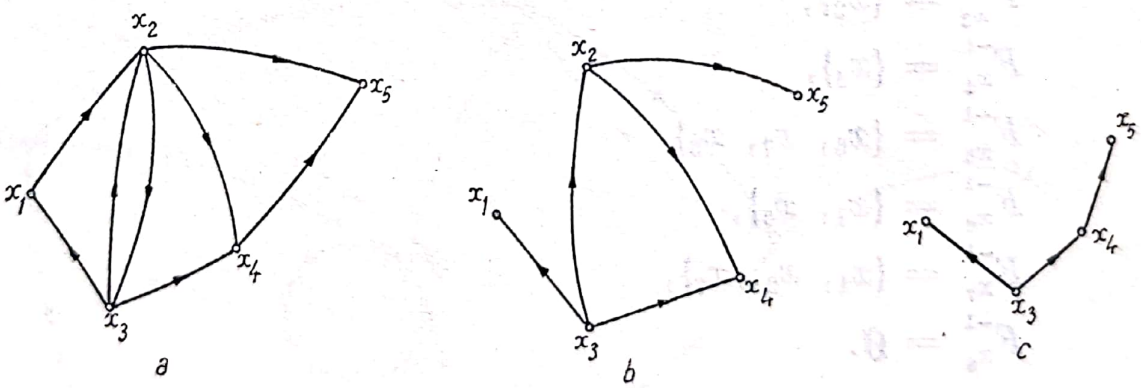


Fig. I.1.7

[1] Prin buclă se înțelege un arc a cărui extremitate inițială coincide cu extremitatea finală.

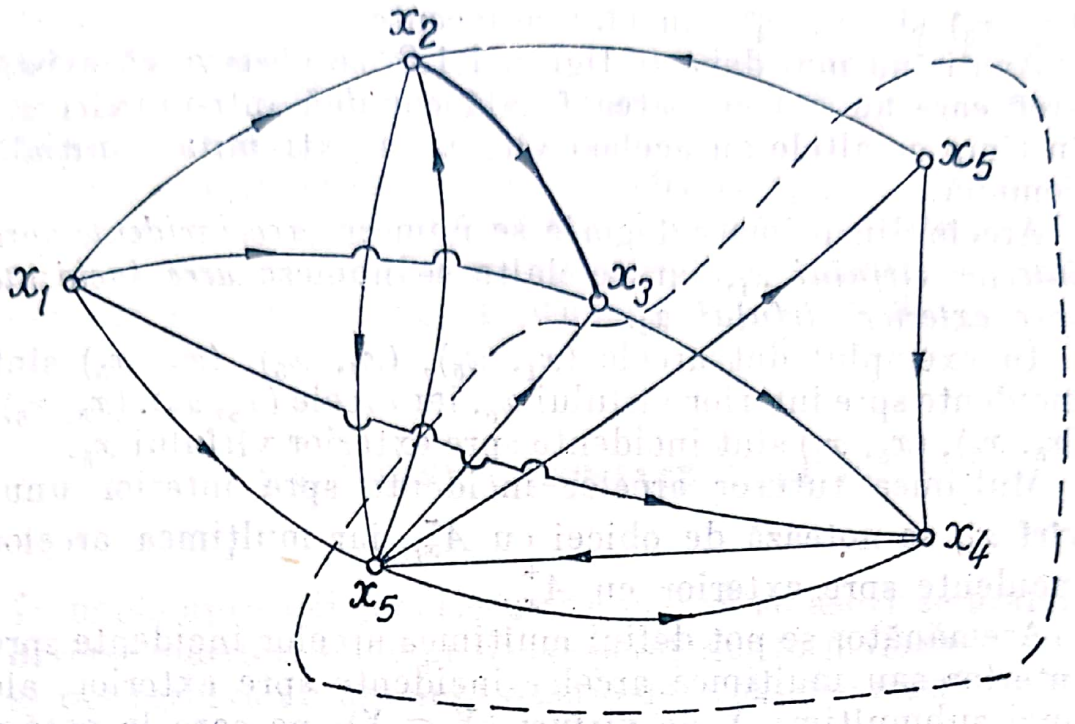


Fig. I.1.8

graful care se obține din graful dat, prin suprimarea unuia sau mai multor arce, cu condiția ca vîrfurile să rămînă aceleași. Dacă $G = (X, A)$ este graful dat, graful parțial dedus din el este

$$G = (X, A^*),$$

unde $A^* \subset A$.

Dacă se consideră graful din figura I.1.7 *a*, atunci în fig. I.1.7 *b* este redat graful parțial corespunzător.

Dacă se suprimă unul sau mai multe vîrfuri și arcele care sosesc sau pleacă din ele, atunci graful obținut se numește *subgraf* al grafului dat. Graful din figura I.1.7 *c* este subgraful celui din figura I.1.7. *a*.

Să considerăm acum graful din figura I.1.8.

Observăm că între vîrfurile x_1 și x_2 sau x_5 și x_2 există cel puțin un arc care le leagă; asemenea vîrfuri se numesc *vîrfuri adiacente*. Vîrfurile x_3 și x_6 de exemplu, nu posedă proprietate de adiacență, neexistînd nici un arc care să le lege.

Arcele care au o extremitate comună se numesc *arce adiacente*. În baza definiției date arcele (x_1, x_2), (x_2, x_3), (x_2, x_5)

sînt adiacente, avînd vîrful x_2 comun, în timp ce arcele (x_2, x_3) și (x_5, x_4), nu sînt adiacente.

Analizînd mai departe figura I.1.8, se observă că există arce care au extremitatea finală comună într-un vîrf x_i, în timp ce altele au același vîrf x_i ca extremitate inițială comună.

Arcele din prima categorie se numesc *arce incidente spre interior vîrfului* x_i, iar celelalte se numesc *arce incidente spre exterior vîrfului* x_i.

În exemplul dat arcele (x_1, x_5), (x_4, x_5), (x_2, x_5) sînt incidente spre interior vîrfului x_5, iar arcele (x_5, x_2), (x_5, x_6), (x_5, x_3), (x_5, x_4) sînt incidente spre exterior vîrfului x_5.

Mulțimea tuturor arcelor incidente spre interior unui vîrf x_i, se notează de obicei cu $A^-_{x_i}$, iar mulțimea arcelor incidente spre exterior cu $A^+_{x_i}$.

Asemănător se pot defini mulțimea arcelor incidente spre interior sau mulțimea arcelor incidente spre exterior, ale unei submulțimi Y de vîrfuri ($Y \subset X$), pe care le notăm respectiv prin A^-_y și A^+_y.

Dacă în figura I.1.8 luăm drept Y submulțimea $Y = (x_4, x_5, x_6)$, atunci vom putea scrie

$$A^-_y = \{(x_1, x_5), (x_1, x_4), (x_3, x_4), (x_2, x_5)\} \quad \text{și}$$

$$A^+_y = \{(x_5, x_2), (x_6, x_2)\}.$$

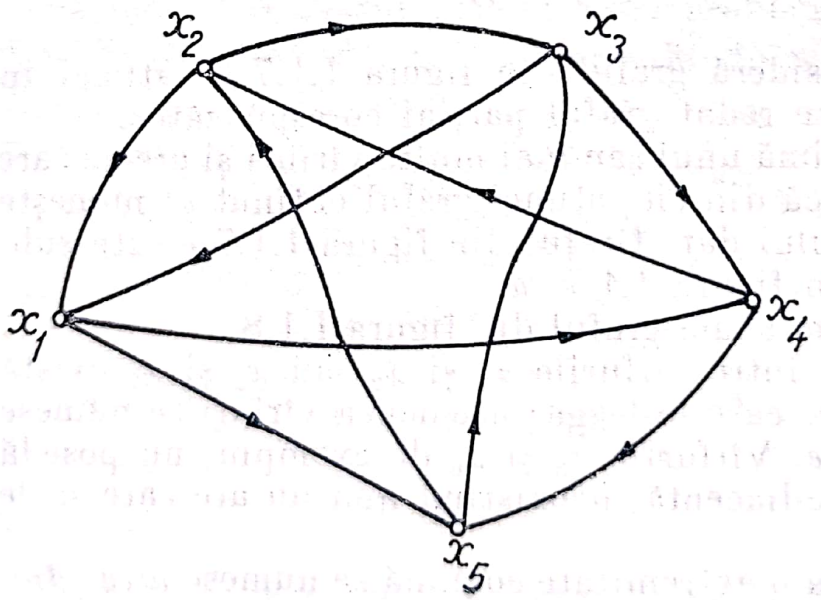


Fig. I.1.9

Scanned with OKEN Scanner

Graful în care, oricare ar fi două vîrfuri x_i și x_j, există cel puțin o muchie care le leagă, se numește *graf complet*. Altfel spus, dacă avem $(x_i, x_j) \in A$, atunci aceasta implică $(x_j, x_i) \in A$. Astfel, graful din figura I.1.9 este un exemplu de graf complet.

Dacă pentru orice pereche de vîrfuri x_i și x_j, existența unui arc între cele două vîrfuri implică existența arcului de sens contrar, graful se numește *graf simetric*. Cu alte cuvinte, dacă $(x_i, x_j) \in A$, atunci și $(x_j, x_i) \in A$.

§ 2. MATRICE ASOCIATE UNUI GRAF

În unele aplicații practice, este util să se asocieze grafelor diverse matrice, unde pe baza operațiilor formale cu acestea se pot rezolva cu ușurință o serie de probleme referitoare la grafe.

Dintre acestea prezentăm următoarele:

1°. **Matricea de incidență a arcelor grafului,** al cărei termen general r_{ij} este definit astfel:

$$r_{ij} = \begin{cases} +1, & \text{dacă arcul } a_j \text{ este incident spre exterior vîrfului } x_i; \\ -1, & \text{dacă arcul } a_j \text{ este incident spre interior vîrfului } x_i; \\ 0, & \text{dacă } x_i \text{ nu este extremitate a arcului } a_j. \end{cases}$$

De exemplu, dacă considerăm graful din figura I.2.1, acestuia îi corespunde următoarea matrice de incidență a arcelor:

$$R = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{cccccccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} \end{array} \\ \left\| \begin{array}{cccccccccc} +1 & 0 & -1 & +1 & 0 & 0 & 0 & 0 & 0 & +1 \\ -1 & +1 & 0 & 0 & 0 & 0 & 0 & +1 & -1 & 0 \\ 0 & -1 & +1 & -1 & -1 & +1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & 0 & +1 & -1 & +1 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \end{array} \right\| \end{array}$$

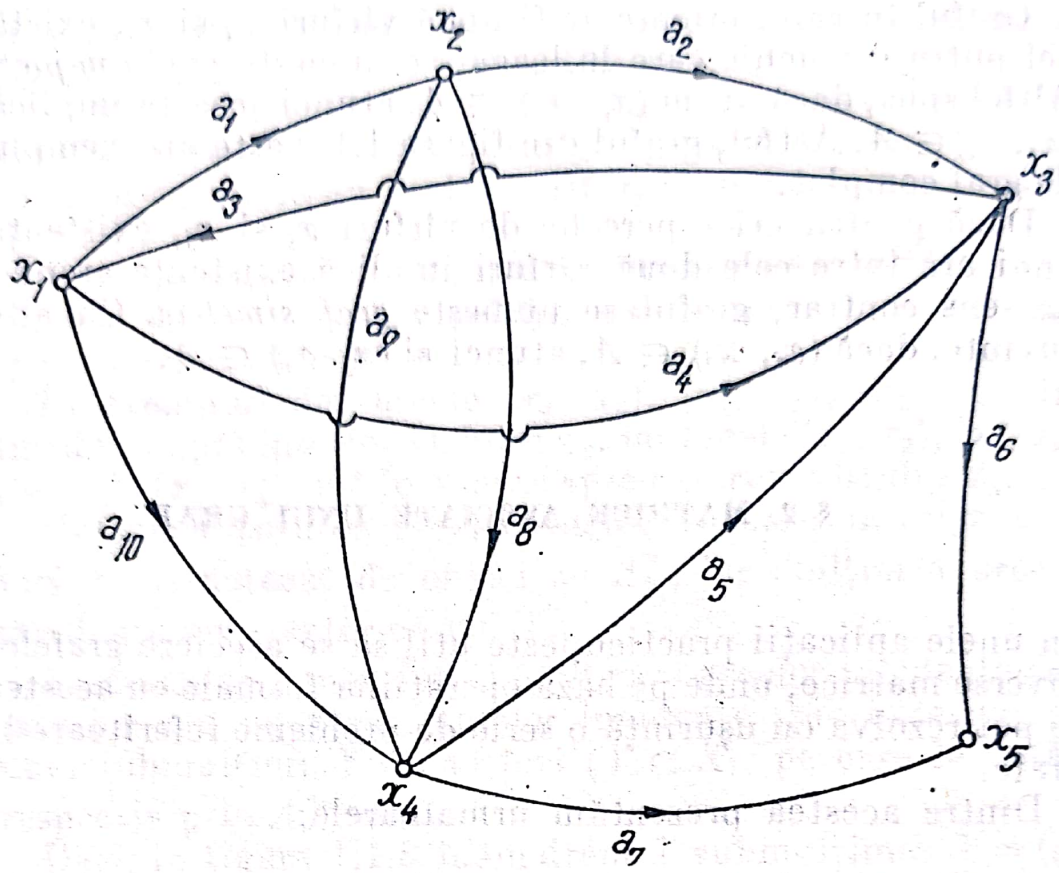


Fig. I.2.1

Numărul elementelor egale cu $+1$ de pe linia vîrfului x_i, ne dă numărul de arce incidente spre exterior, vîrfului x_i, în timp ce numărul elementelor egale cu -1 ne dă numărul de arce incidente spre interior aceluiași vîrf.

2°. Matrice pătrate asociate unui graf cu elementele

$$b_{ij} = \begin{cases} 1, \text{ dacă există un arc ce leagă vîrful } x_i \text{ cu vîrful } x_j, \\ 0, \text{ dacă nu există un asemenea arc.} \end{cases}$$

Matricea $B = \| b_{ij} \|$, $i, j, = 1, \ldots, n$, ale cărei elemente sînt egale cu 1 și 0, se numește *matrice booleană* asociată grafului $G = (X, F)$.

Astfel, grafului din figura I.2.1 îi corespunde matricea booleană:

$$B = \begin{array}{c|ccccc} & x_1 & x_2 & x_3 & x_4 & x_5 \\ \hline x_1 & 0 & 1 & 0 & 1 & 0 \\ x_2 & 0 & 0 & 1 & 1 & 0 \\ x_3 & 1 & 0 & 0 & 0 & 1 \\ x_4 & 0 & 1 & 1 & 0 & 1 \\ x_5 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Dacă în matricea B, astfel construită, elementele egale cu unu le înlocuim prin zero, iar în locul elementelor nule punem unu, atunci obținem *complementara* matricei B, pe care o notăm cu B'. Matricei B' îi corespunde un graf $G' = (X, F')$, având aceeași mulțime de vîrfuri X, numit

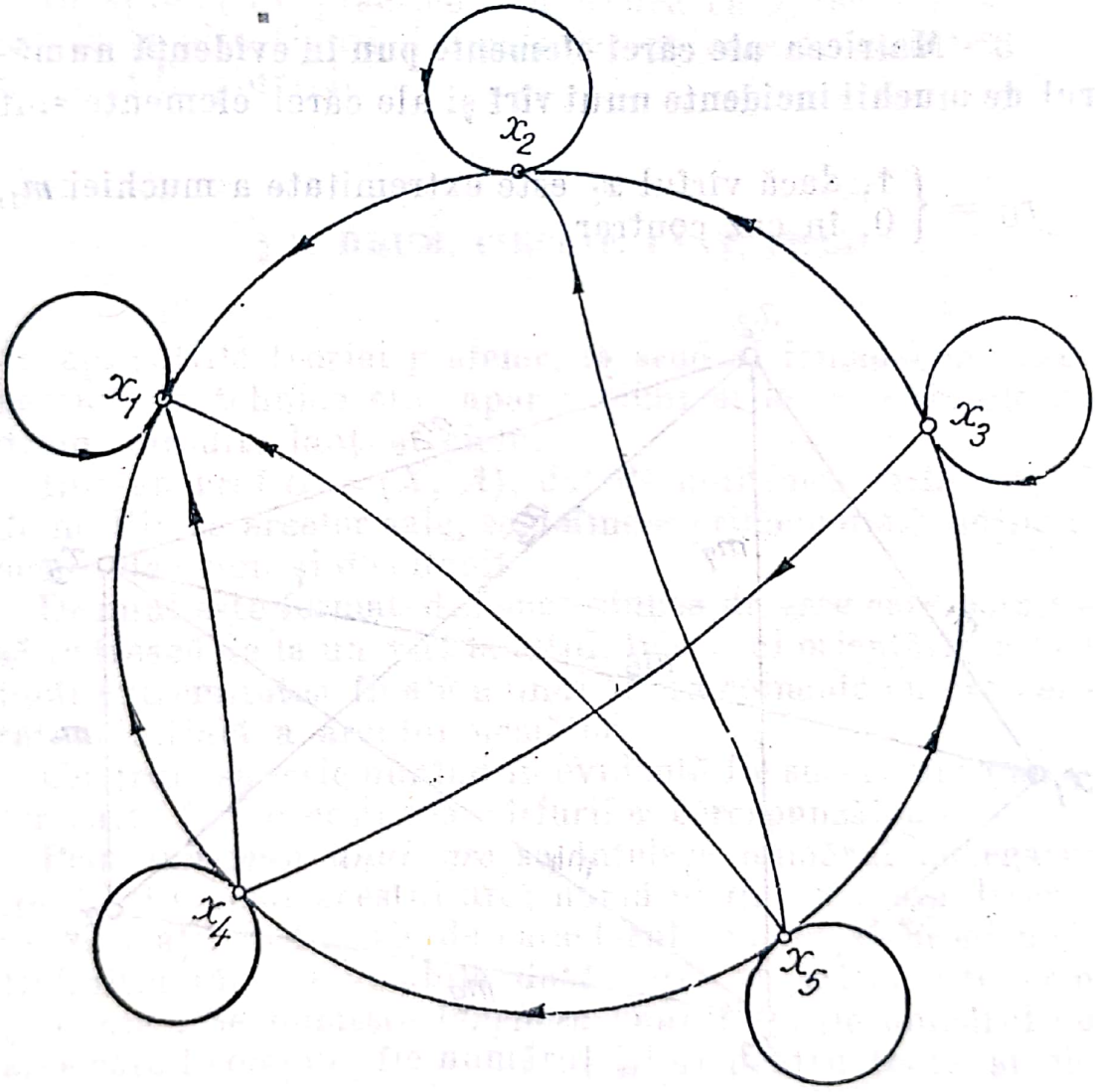


Fig. I.2.2

graf complementar grafului G. Astfel, complementara matricei grafului din figura I.2.1 este

$$B' = \begin{array}{c|ccccc} & x_1 & x_2 & x_3 & x_4 & x_5 \\ \hline x_1 & 1 & 0 & 1 & 0 & 1 \\ x_2 & 1 & 1 & 0 & 0 & 1 \\ x_3 & 0 & 1 & 1 & 1 & 0 \\ x_4 & 1 & 0 & 0 & 1 & 0 \\ x_5 & 1 & 1 & 1 & 1 & 1 \end{array}$$

iar graful complementar, corespunzător acestei matrici, este cel din figura I.2.2.

3°. **Matricea ale cărei elemente pun în evidență numărul de muchii incidente unui vîrf** și ale cărei elemente sînt

$$r_{ij} = \begin{cases} 1, \text{ dacă vîrful } x_i \text{ este extremitate a muchiei } m_j, \\ 0, \text{ în caz contrar.} \end{cases}$$

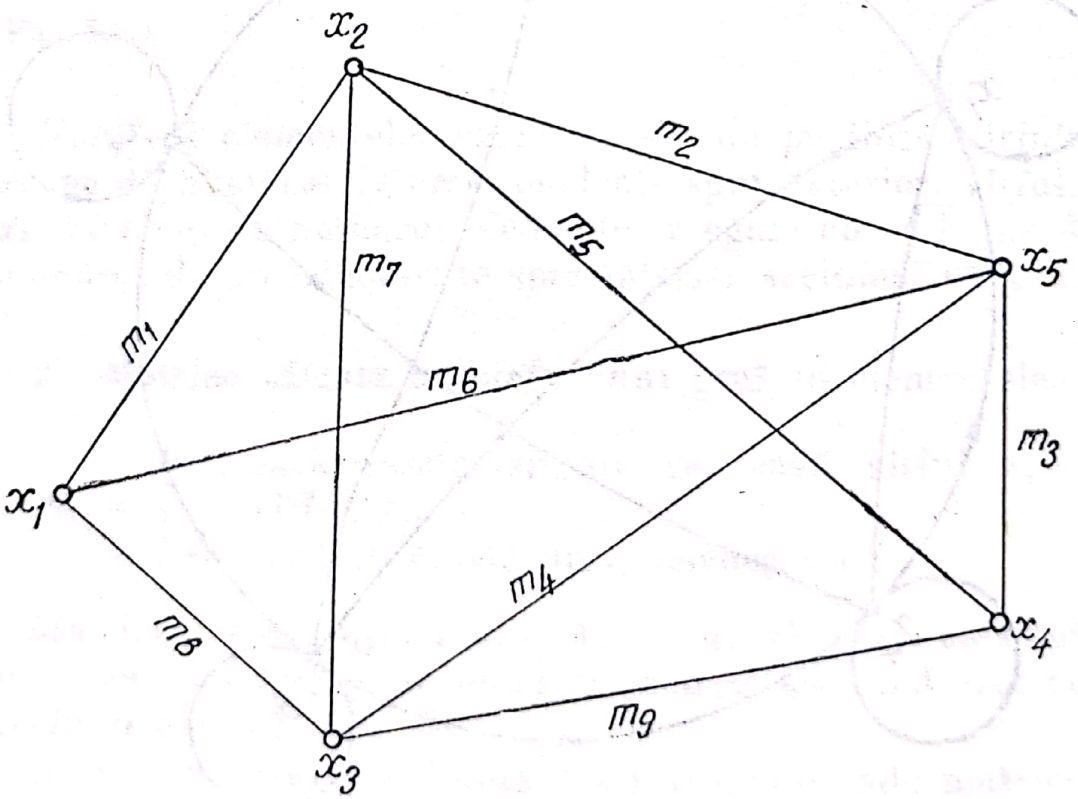


Fig. I.2.3

Scanned with OKEN Scanner

Fie graful din figura I.2.3. Avem matricea

$$M = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{ccccccccc} m_1 & m_2 & m_3 & m_4 & m_5 & m_6 & m_7 & m_8 & m_9 \end{array} \\ \left\| \begin{array}{ccccccccc} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{array} \right\| \end{array}$$

Numărul elementelor de pe linia (i), egale cu unitatea, ne dă numărul muchiilor incidente vîrfului $x_i \in X$, $i = 1, 2, \ldots, 9$.

În aplicațiile practice vom vedea că grafelor li se pot asocia și alte tipuri de matrice, pe care le vom trata la momentul potrivit.

§ 3. DRUM, CIRCUIT, LANȚ, CICLU

În aplicațiile teoriei grafelor, la studiul fenomenelor economice, în tehnică etc. apar noțiuni utile ca noțiunile de drum, circuit, lanț și ciclu.

Într-un graf $G = (X, A)$, dat de mulțimea vîrfurilor și de mulțimea arcelor sale, se definesc primele două noțiuni, acelea de drum și de circuit.

Drumul este format din succesiunea de arce care permite să se treacă de la un vîrf la altul, în sensul orientării, astfel încît extremitatea finală a unui arc să coincidă cu extremitatea inițială a arcului următor.

Un drum se scrie punînd în evidență fie succesiunea arcelor sale, fie succesiunea vîrfurilor corespunzătoare.

Prin *lungimea unui arc* se înțelege numărul nenegativ $l(a) \geqslant 0$ asociat acestui arc; numărul $l(a) \geqslant 0$ are diverse semnificații în funcție de caracterul concret al problemei. Definiția rămîne valabilă dacă „arc" se înlocuiește prin „muchie". Se numește *lungimea unui drum* fie numărul de arce care-l compun, fie numărul $\Sigma l(a)$ pentru toate arcele drumului.

Scanned with OKEN Scanner

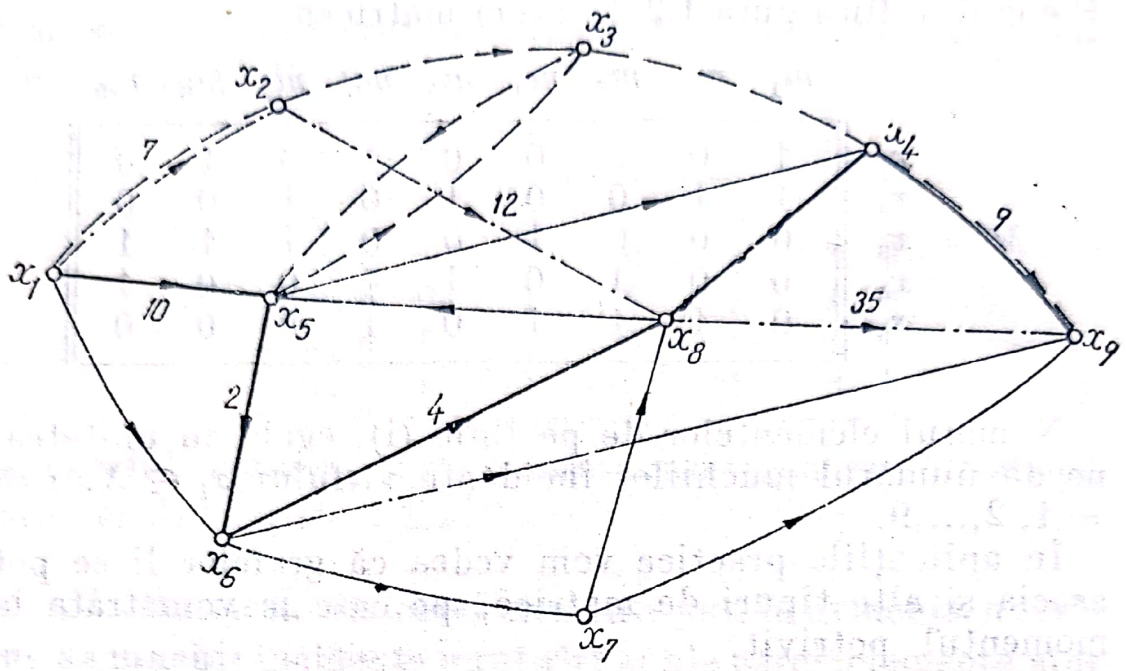


Fig. I.3.1

Pe graful din figura I.3.1 se pot citi de exemplu drumurile: $d^{(1)} = d(x_1, x_9) = \{x_1, x_5, x_6, x_8, x_4, x_9\}$ pus în evidență de vîrfurile sale, sau $d^{(1)} = d(x_1, x_9) = \{(x_1, x_5), (x_5, x_6), (x_6, x_8), (x_8, x_4), (x_4, x_9)\}$ pus în evidență de arcele sale.

Conform definiției, lungimea acestui drum este

$$l(d^{(1)}) = 5.$$

$$d^{(2)} = d(x_1, x_9) = \{x_1, x_2, x_8, x_9\}$$

sau

$$d^{(2)} = \{(x_1, x_2), (x_2, x_8), (x_8, x_9)\}$$

cu o lungime

$$l(d^{(2)}) = 3$$

sau

$$l(d^{(1)}) = 10 + 2 + 4 + 6 + 9 = 31$$

și

$$l(d^{(2)}) = 7 + 12 + 35 = 54.$$

Observații:

Se constată că drumurile $d^{(1)}$ și $d^{(2)}$ folosesc o dată și numai o dată fiecare vîrf al său, în timp ce drumul

$$d^{(3)} = \{x_1, x_2, x_3, x_5, x_3, x_4, x_9\}$$

folosește de două ori vîrful x_3.

Scanned with OKEN Scanner

Pentru aceste considerente drumurile $d^{(1)}$ și $d^{(2)}$ se numesc *elementare* iar drumul $d^{(3)}$ *neelementar*.

Drumul particular, care trece prin toate vîrfurile grafului se numește drum *hamiltonian*.

Pe graful din figura I.3.2 am trasat cu linie îngroșată drumul hamiltonian.

$$d_h = \{x_1, x_2, x_3, x_4\}$$

Dacă extremitatea inițială a unui drum finit coincide cu extremitatea sa finală, drumul poartă denumirea de *circuit*.

Dacă circuitul conține un singur arc, se spune că formează o *buclă*. De exemplu, în fig. I.3.2 în vîrful x_3 există o buclă.

Într-un mod asemănător se definesc noțiunile de lanț și ciclu.

Lanțul este o succesiune de muchii, în care extremitatea terminală a uneia, coincide cu extremitatea inițială a muchiei următoare.

Un lanț închis, adică un lanț în care extremitatea inițială coincide cu extremitatea finală, se numește *ciclu*.

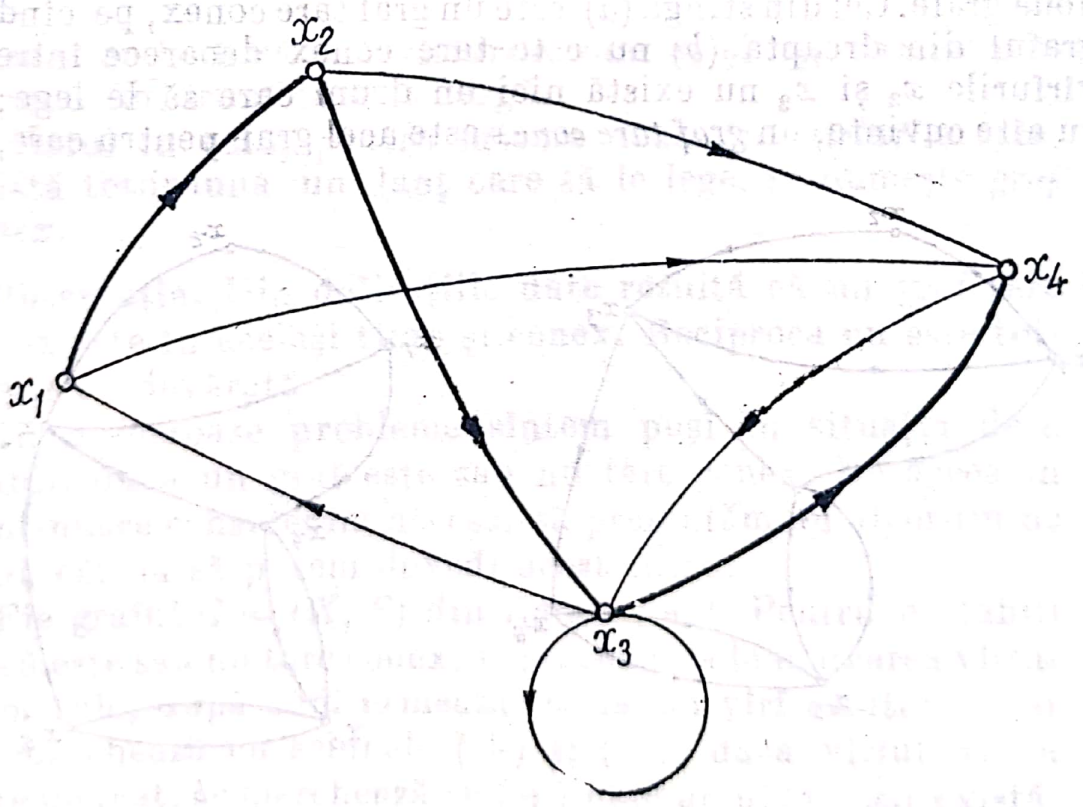


Fig. I.3.2

Scanned with OKEN Scanner

De exemplu în figura 1.3.2 avem ciclul $\{x_1, x_2, x_4, x_1\}$.

Cititorul constată cu uşurinţă că noţiunile de drum şi circuit impun o anumită orientare, în timp ce noţiunile de lanţ şi ciclu, nu impun orientare.

§ 4. GRAF TARE CONEX. GRAF CONEX. COMPONENTĂ CONEXĂ A GRAFULUI. ARBORI ÎNTR-UN GRAF

1°. Noţiunea de conexiune a unui graf este foarte importantă în teoria grafelor şi mai ales în aplicaţiile ei.

Este suficient să ne imaginăm un graf care ar reprezenta posibilităţile de lansare în producţie a unor repere. Oricare ar fi două repere x_i, x_k, există totdeauna o succesiune de repere care să conducă de la x_i la x_k, sau invers. Graful corespunzător unei asemenea probleme este un graf tare conex. Pentru exemplificare, în figura I.4.1 sînt redate două grafe. Cel din stînga (a) este un graf tare conex, pe cînd graful din dreapta (b) nu este tare conex deoarece între vîrfurile x_3 şi x_2 nu există nici un drum care să le lege; cu alte cuvinte, un *graf tare conex* este acel graf pentru care,

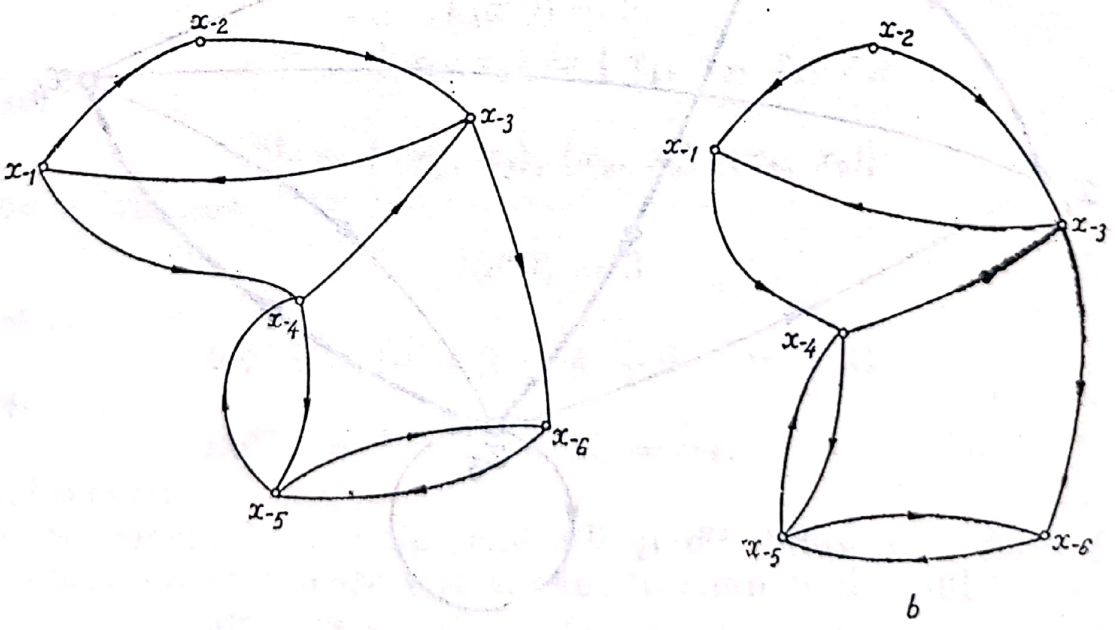


Fig. I.4.1

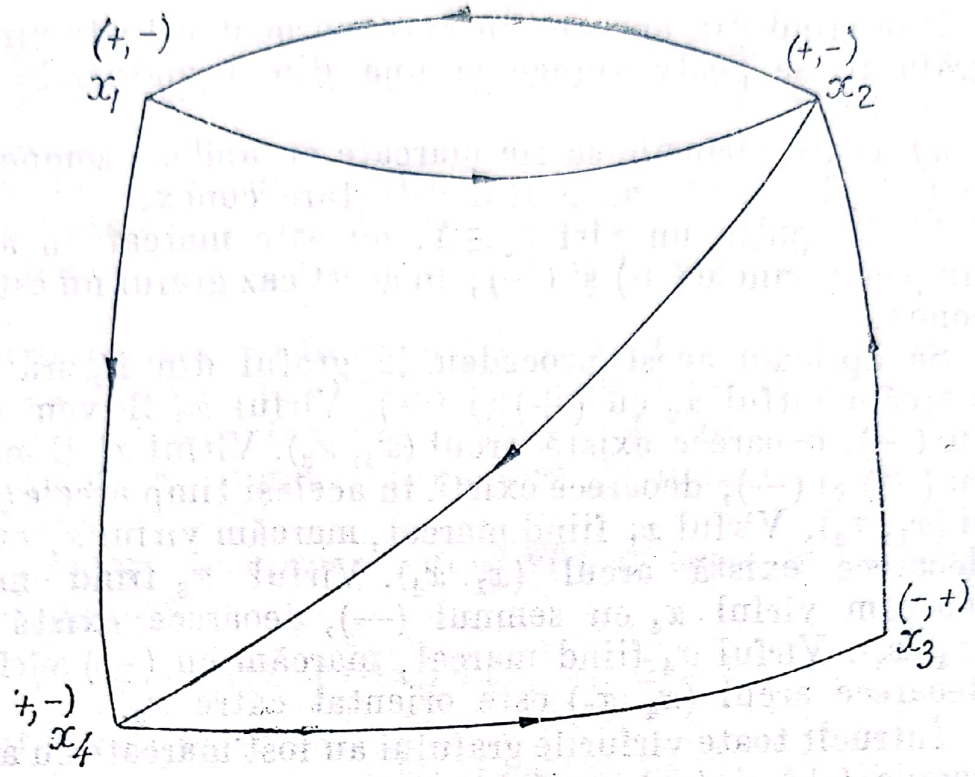


Fig. I.4.2

oricare ar fi două din vîrfurile sale x_k și x_q, există totdeauna un drum care le leagă.

Graful în care, pentru oricare două din vîrfurile sale, există totdeauna un lanț care să le lege, se numește *graf conex*.

Observație. Din definițiile date rezultă că un graf tare conex este în același timp și conex. Reciproca nu este totdeauna adevărată.

În numeroase probleme sîntem puși în situația de a stabili dacă un graf este sau nu tare conex. De aceea în continuare considerăm necesar să prezentăm un algoritm pe baza căruia să putem dovedi acest lucru.

Fie graful $G = (X, F)$ din figura I.4.2. Pentru a stabili dacă este sau nu tare conex, se procedează la marcarea vîrfurilor sale, după cum urmează: se ia un vîrf arbitrar x_i și se marchează cu semnele (+) și (—); dacă vîrful x_j nu este marcat, se marchează cu (+) cînd arcul (x_i, x_j) există, și cu (—) dacă arcul (x_j, x_i) există.

Scanned with OKEN Scanner

Procedînd din aproape în aproape pentru toate vîrfurile grafului, se poate ajunge la una din următoarele două situații:

a) Toate vîrfurile să fie marcate cu ambele semne, (+) și (—); în acest caz graful este tare conex.

b) Cel puțin un vîrf $x_i \in X$, nu este marcat în același timp cu semnele (+) și (—); în acest caz graful nu este tare conex.

Să aplicăm acest procedeu la graful din figura I.4.1. Marcăm vîrful x_2 cu (+) și (—). Vîrful x_3 îl vom marca cu (—), deoarece există arcul (x_3, x_2). Vîrful x_1 îl marcăm cu (+) și (—), deoarece există în același timp arcele (x_2, x_1) și (x_1, x_2). Vîrful x_1 fiind marcat, marcăm vîrful x_4 cu (+), deoarece există arcul (x_1, x_4). Vîrful x_3 fiind marcat, marcăm vîrful x_4 cu semnul (—), deoarece există arcul (x_4, x_3). Vîrful x_4 fiind marcat, marcăm cu (+) vîrful x_3, deoarece arcul (x_4, x_3) este orientat către x_3.

Întrucît toate vîrfurile grafului au fost marcate cu ambele semne (+) și (—), graful este tare conex.

2°. Nu puține sînt situațiile în care se ridică probleme de aprovizionare a mai multor centre de la unul și același depozit. Astfel, fiind dat un depozit central cu o anumită marfă în stoc, se cere să se aprovizioneze diverse centre de consum, cu marfa respectivă. Cunoscîndu-se distanțele dintre depozitul central și diferitele centre consumatoare, se pune problema găsirii unui drum de lungime minimă, dar care să treacă prin toate centrele de consum.

Asociind problemei date graful corespunzător, drumul căutat va fi arborele de lungime minimă al grafului respectiv.

Prin urmare, *arborele unui graf* este un graf parțial al grafului dat, conex și fără cicluri.

Dată fiind importanța noțiunii de arbore în problemele de aprovizionare, precum și lipsa de circuite a acestuia, este necesar să dispunem de un procedeu, care să ne asigure de lipsa circuitelor într-un graf, mai ales cînd acesta posedă un număr mare de vîrfuri.

În acest caz se folosește din aproape în aproape procedeul marcării vîrfurilor grafului după cum urmează:

a) Se marchează cu semnul (+) un vîrf $x_i \in X$ pentru care $F_{x_i} = \emptyset$ (nu are succesori).

Scanned with OKEN Scanner

b) Vîrful x_i fiind marcat, se marchează cu $(+)$ oricare din vîrfurile $x_j \in X$ pentru care vîrfurile F_{x_j} au fost deja marcate.

Dacă prin procedeul descris toate vîrfurile grafului au fost marcate cu $(+)$, graful este fără circuite. Să arătăm că un asemenea graf nu conține decît drumuri de lungime finită.

Pentru aceasta, vîrfului inițial marcat, x_i, să-i asociem lungimea $L(x_i) = 0$; vîrfului x_j, marcat în etapa următoare, îi asociem lungimea $L(x_j) = 1$, ceea ce exprimă existența unor drumuri elementare de lungime unu care pleacă din x_j.

Din aproape în aproape, unui vîrf x_k oarecare i se asociază lungimea

$$L(x_k) = \max \{ L(x_j) \mid (x_k, x_j) \in A \} + 1;$$

în felul acesta se va găsi că

$$L(x_k) < +\infty$$

oricare ar fi $x_k \in X$.

Pentru exemplificare să considerăm graful din figura I.4.3. Se constată că vîrful x_3 nu are succesori, adică $F_{x_3} = \emptyset$; deci îl marcăm cu $(+)$. Vîrful x_3 fiind marcat, marcăm

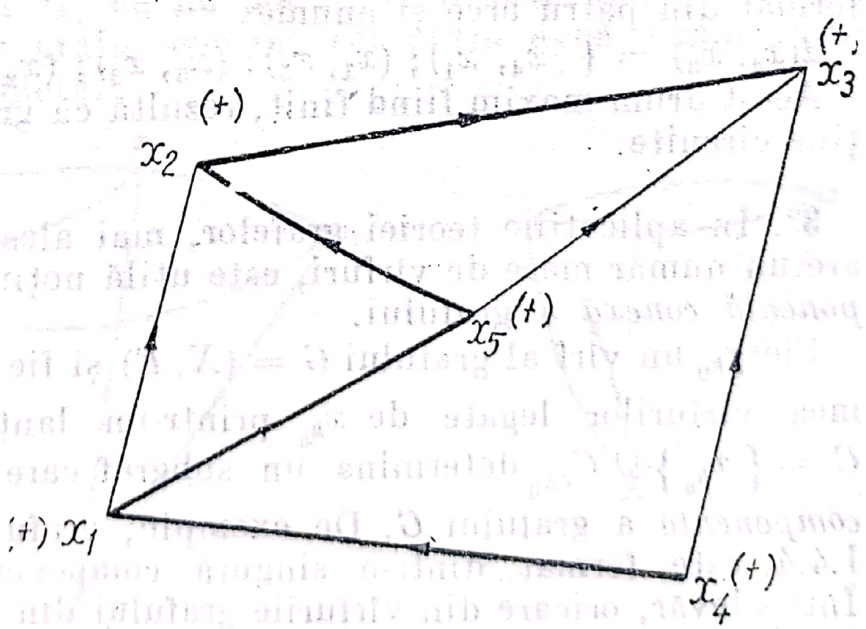


Fig. I.4.3

vîrfurile x_2 și x_4 cu semnul (+), deoarece au ca succesori numai vîrful x_3 deja marcat. Vîrful x_5 îl vom marca cu (+), întrucît are ca succesori vîrfurile x_2 și x_3 deja marcate. În fine, vîrful x_1 îl vom marca cu (+), fiindcă succesorii vîrfului sînt deja marcați.

Întrucît toate vîrfurile grafului au fost marcate, graful nu are circuite.

În baza celor arătate, vîrfului x_3, cu $F_{x_3} = \emptyset$, îi atașăm lungimea $L(x_3) = 0$.

Vîrfurile x_2, cu $F_{x_2} = x_3$, îi asociem lungimea $L(x_2) = \max \{L(x_i)/(x_2, x_i) \in A\} + 1 = \max \{0\} + 1 = 1$.

Vîrfului x_5, cu $F_{x_5} = \{x_2, x_3\}$, îi asociem lungimea $L(x_5) = \max \{L(x_i)/(x_5, x_i) \in A\} + 1 = \max \{L(x_2), + + L(x_3)\} + 1 = \max \{1, 0\} + 1 = 2$.

Vîrfului x_1, cu $F_{x_1} = \{x_2, x_5\}$, îi asociem lungimea $L(x_1) = \max \{L(x_i)/(x_1, x_i) \in A\} + 1 = \max \{L(x_2), L(x_5)\} + + 1 = \max \{1, 2\} + 1 = 3$.

Vîrfului x_4, cu $F_{x_4} = \{x_1, x_3\}$, îi asociem lungimea $L(x_4) = \max \{L(x_i)/(x_4, x_i) \in A\} + 1 = \max \{L(x_1), L(x_3)\} + 1 = \max \{3, 0,\} + 1 = 4$.

Drumul de lungime maximă, în graful considerat, este format din patru arce și anume

$d(x_4, x_3) = \{(x_4, x_1); (x_1, x_5); (x_5, x_2); (x_2, x_3)\}$

Acest drum maxim fiind finit, rezultă că graful nu conține circuite.

3°. În aplicațiile teoriei grafelor, mai ales cînd graful are un număr mare de vîrfuri, este utilă noțiunea de *componentă conexă* a grafului.

Fie x_{k_0} un vîrf al grafului $G = (X, F)$ și fie $C_{x_{k_0}}$ mulțimea vîrfurilor legate de x_{k_0} printr-un lanț. Mulțimea $C = \{x_{k_0}\} \cup C_{x_{k_0}}$ determina un subgraf care se numește *componentă* a grafului G. De exemplu, graful din figura I.4.4. este format dintr-o singură componentă conexă. Într-adevăr, oricare din vîrfurile grafului din figură poate fi legat printr-un lanț de toate celelalte vîrfuri ale grafului.

Scanned with OKEN Scanner

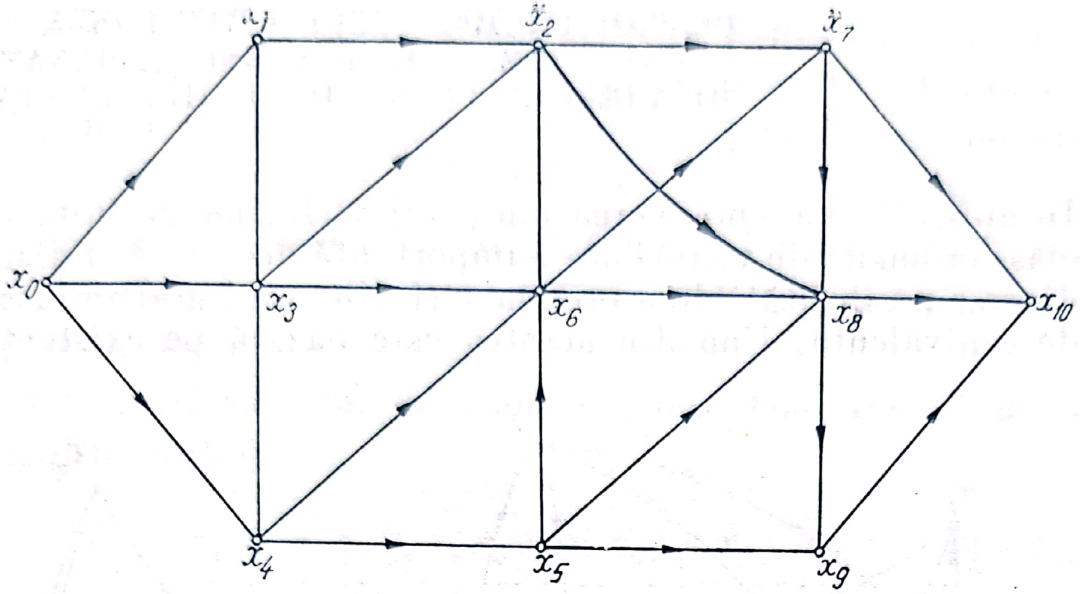


Fig. I.4.4

Graful din figura I.4.5 reprezintă două componente conexe distincte:

$Y_1 = \{ x_1, x_2, x_3, x_4 \}$ și $Y_2 = \{ x_5, x_6, x_7 \}$.

Observații:

a) Dacă graful este conex, el nu are decît o singură componentă conexă.

b) Dacă graful nu este conex, există cel puțin două vîrfuri, x_i și x_k, ce nu pot fi legate prin nici un lanț și în acest caz graful conține cel puțin două componente conexe, distincte.

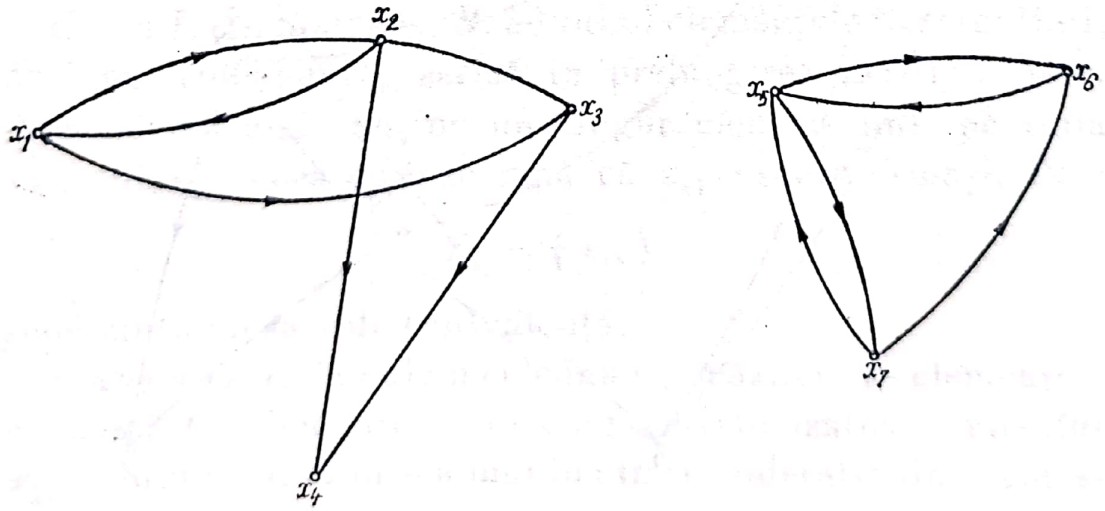


Fig. I.4.5

Scanned with OKEN Scanner

§ 5. DESCOMPUNEREA UNUI GRAF CONEX ȘI FĂRĂ CIRCUITE, ÎN CLASE ORDONATE DE VÎRFURI (CLASE DE ECHIVALENȚĂ)

În aplicații descompunerea mulțimii vîrfurilor grafului în clase ordonate de vîrfuri are o importanță deosebită. Există diverse posibilități de a ordona vîrfurile unui graf în clase de echivalență. Una din acestea este bazată pe existența

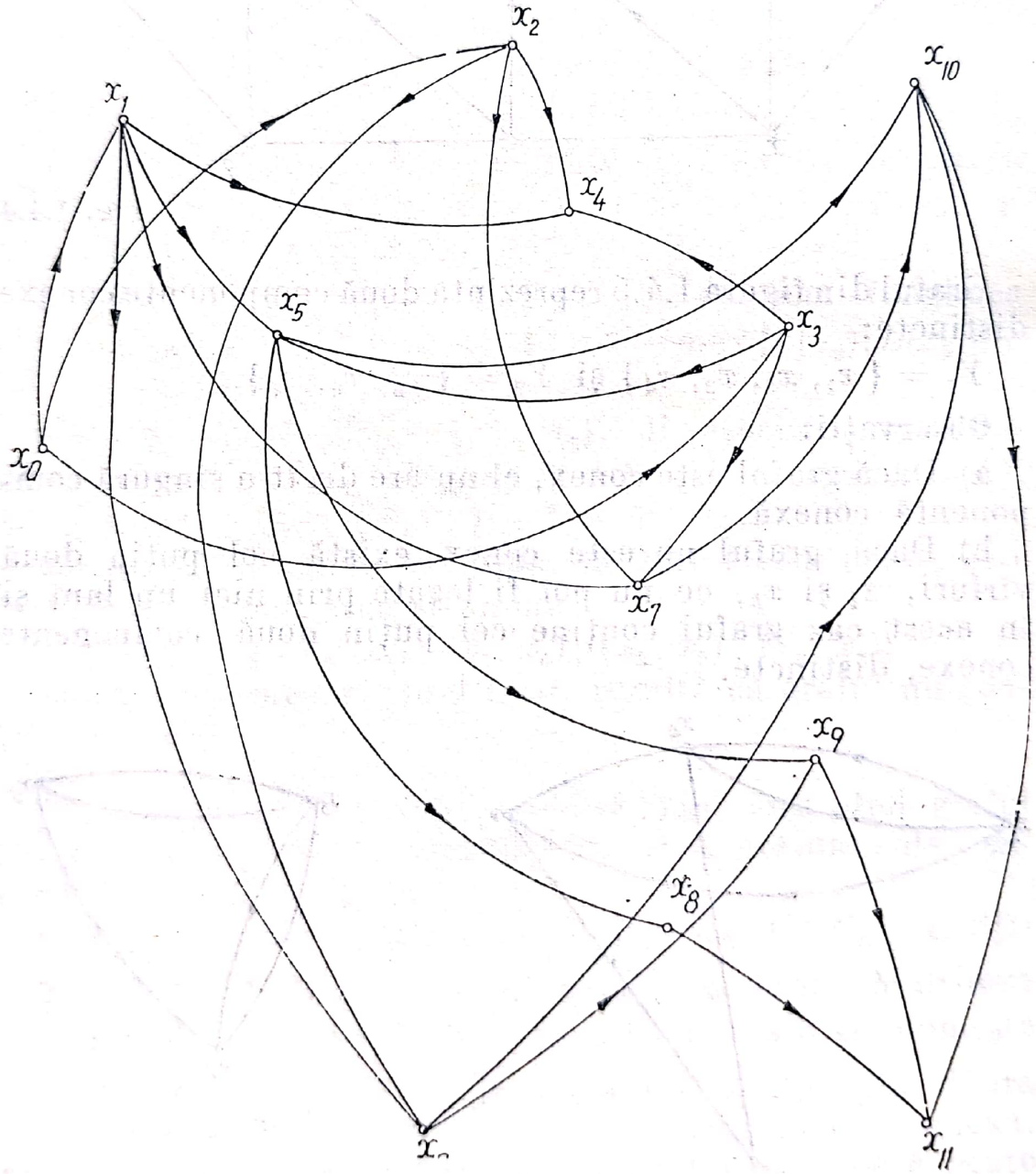


Fig. I.5.1

urmaşilor, de un anumit ordin, ai vîrfurilor grafului. Astfel, vom spune că vîrful x_i face parte din clasa de echivalenţă Y_k, dacă vîrful x_i are urmaşi de ordinul k, dar nu are urmaşi de ordinul $(k+1)$, adică $F^k_{x_i} \neq \emptyset$ şi $F^{k+1}_i = \emptyset$.

Să arătăm, pentru cazul grafului din figura I.5.1, cum se procedează practic la împărţirea mulţimii vîrfurilor sale în asemenea clase.

Mai întîi asociem grafului matricea booleană B, după regulile arătate.

$B =$

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	C_0	C_1	C_2	C_3	C_4
x_0	0	1	1	1	0	0	0	0	0	0	0	0	3	3	3	3	0
x_1	0	0	0	0	1	0	1	1	0	0	0	0	3	3	3	0	—
x_2	0	0	0	0	1	0	1	1	0	0	0	0	3	3	3	0	—
x_3	0	0	0	0	1	1	0	1	0	0	0	0	3	3	3	0	—
x_4	0	0	0	0	0	0	0	0	1	1	0	0	2	2	0	—	—
x_5	0	0	0	0	0	0	0	0	1	1	1	0	3	3	0	—	—
x_6	0	0	0	0	0	0	0	0	0	1	1	0	2	2	0	—	—
x_7	0	0	0	0	0	0	0	0	0	1	1	0	2	2	0	—	—
x_8	0	0	0	0	0	0	0	0	0	0	0	1	1	0	—	—	—
x_9	0	0	0	0	0	0	0	0	0	0	0	1	1	0	—	—	—
x_{10}	0	0	0	0	0	0	0	0	0	0	0	1	1	0	—	—	—
x_{11}	0	0	0	0	0	0	0	0	0	0	0	0	0	—	—	—	—

Etapa I. În matricea B, adunînd elementele fiecărei linii, obţinem coloana C_0, scrisă în prelungirea matricei date. Această coloană conţine un singur element nul, pe linia vîrfului x_{11}, ceea ce înseamnă că x_{11} nu are urmaşi. Deci

$$Y_1 = \{x_{11}\}$$

reprezintă clasa I de echivalenţă.

Etapa a II-a. Calculăm coloana C_1 scăzînd din elementele coloanei C_0 elementele coloanei corespunzătoare vîrfului x_{11}, eliminat, fără însă a mai lua în consideraţie linia corespunzătoare vîrfului x_{11}.

Scanned with OKEN Scanner

Coloana C_1 conține trei zerouri situate pe liniile corespunzătoare vîrfurilor x_8, x_9 și x_{10}. Aceasta înseamnă că în graful rămas, prin suprimarea vîrfului x_{11}. aceste vîrfuri nu au urmași și deci ele formează clasa a II-a de echivalență, și anume

$$Y_2 = \{ x_8, x_9, x_{10} \}$$

Analog formăm coloana C_2, scăzînd din coloana C_1 suma elementelor coloanelor corespunzătoare vîrfurilor eliminate x_8, x_9 x_{10}, obținînd în acest fel clasa a III-a de echivalență

$$Y_3 = \{ x_4, x_5, x_6, x_7 \}.$$

La fel obținem coloana C_3 și clasa a IV-a de echivalență,

$$Y_4 = \{ x_1, x_2, x_3 \}.$$

În sfîrșit, coloana C_4 dă clasa a V-a de echivalență

$$Y_5 = \{ x_0 \}.$$

Graful obținut prin ordonarea vîrfurilor în clasele de echivalență este redat în figura I.5.2.

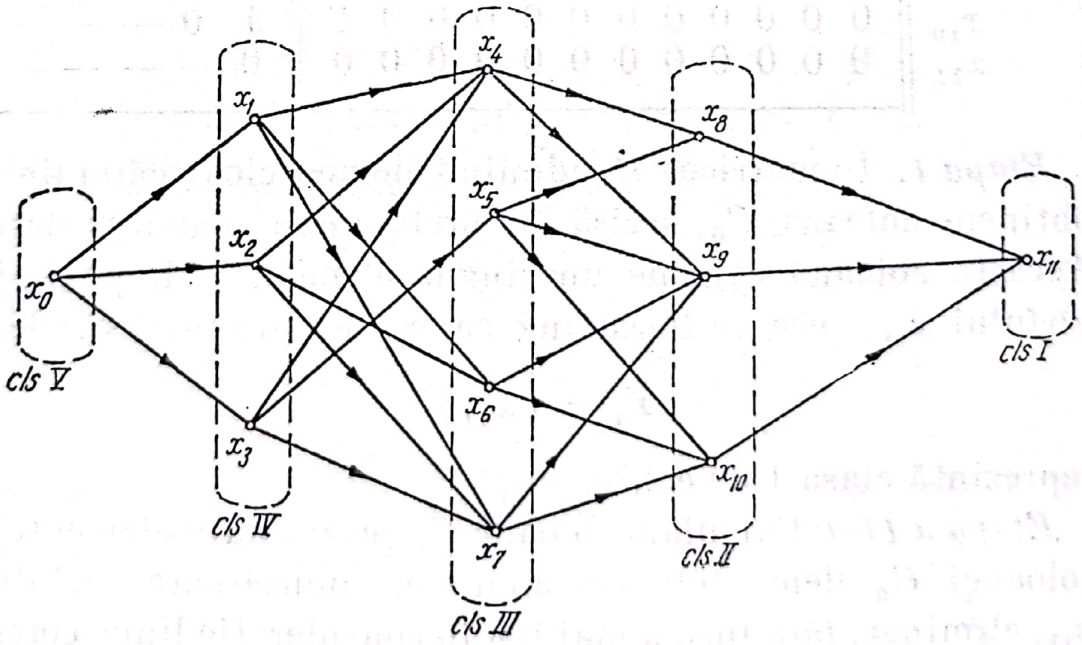


Fig. I.5.2

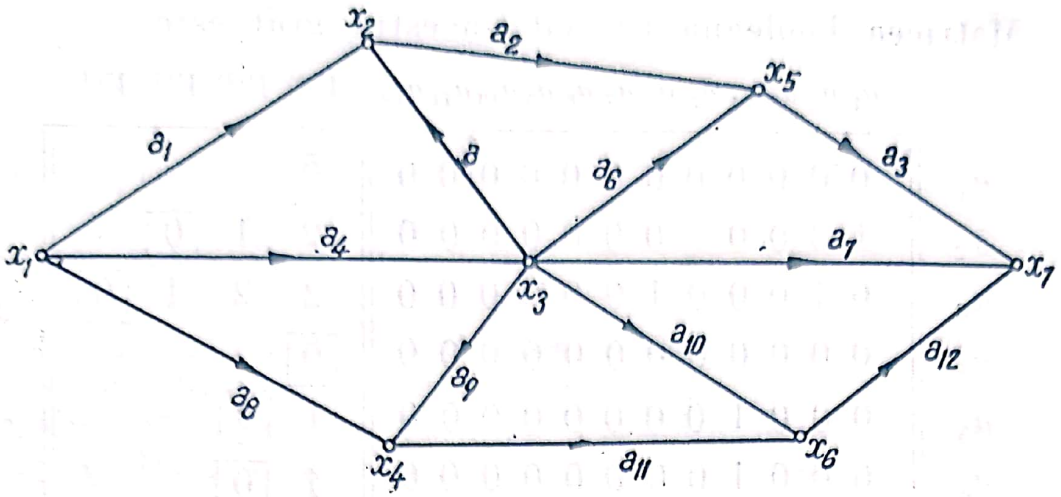


Fig. I.5.3

Metoda arătată sugerează o cale directă de a ordona mulțimea de vîrfuri a grafului, fără a mai folosi matricea booleană asociată grafului și anume:

a) Se consideră vîrfurile x_i pentru care $F_{x_i} = \emptyset$ ca făcînd parte din prima clasă.

Eliminînd în graful dat arcele incidente spre interior vîrfurilor din prima clasă, considerăm în graful rămas vîrfurile x_j fără urmași; ele formează clasa următoare de echivalență.

Procedeul continuă pînă cînd nu mai sînt vîrfuri de luat în considerație.

Într-un mod asemănător se poate face, la nevoie, o ordonare a arcelor grafului.

De această dată matricea A de incidență a vîrfurilor va fi înlocuită cu o nouă matrice, de incidență a arcelor.

Elementele ei r_{ij} se calculează după regula

$$r_{ij} = \begin{cases} 1, \text{ dacă arcul } a_i \text{ pleacă din extremitatea finală} \\ \quad \text{a arcului } a_j; \\ 0, \text{ în caz contrar.} \end{cases}$$

Pentru exemplificare, să luăm graful din figura I.5.3.

Scanned with OKEN Scanner

Matricea booleană asociată acestui graf este

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	$V^{(0)}$	$V^{(1)}$	$V^{(2)}$	$V^{(3)}$
a_1	0	0	0	0	0	0	0	0	0	0	0	0	$\boxed{0}$	—	—	—
a_2	1	0	0	0	1	0	0	0	0	0	0	0	2	1	$\boxed{0}$	—
a_3	0	1	0	0	0	1	0	0	0	0	0	0	2	2	1	$\boxed{0}$
a_4	0	0	0	0	0	0	0	0	0	0	0	0	$\boxed{0}$	—	—	—
a_5	0	0	0	1	0	0	0	0	0	0	0	0	1	$\boxed{0}$	—	—
a_6	0	0	0	1	0	0	0	0	0	0	0	0	1	$\boxed{0}$	—	—
a_7	0	0	0	1	0	0	0	0	0	0	0	0	1	$\boxed{0}$	—	—
a_8	0	0	0	0	0	0	0	0	0	0	0	0	$\boxed{0}$	—	—	—
a_9	0	0	0	1	0	0	0	0	0	0	0	0	1	$\boxed{0}$	—	—
a_{10}	0	0	0	1	0	0	0	0	0	0	0	0	1	$\boxed{0}$	—	—
a_{11}	0	0	0	0	0	0	0	1	1	0	0	0	2	1	$\boxed{0}$	—
a_{12}	0	0	0	0	0	0	0	0	1	1	0	0	2	2	1	$\boxed{0}$

Clasa I	Clasa a II-a	Clasa a III-a	Clasa a IV-a
a_1	a_5	a_2	a_3
a_4	a_6	a_{11}	a_{12}
a_8	a_7		
	a_9		
	a_{10}		

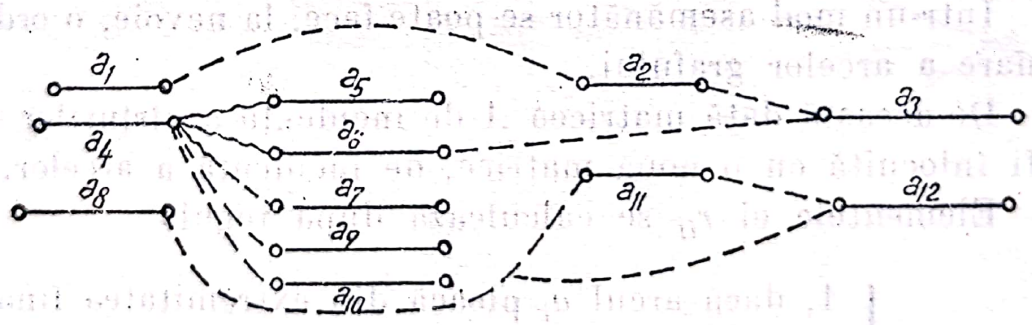


Fig. I.5.4

Pentru ordonarea arcelor în clase, s-au făcut aceleași operații ca și la ordonarea vîrfurilor.

§ 6. NUMERELE DE STABILITATE INTERNĂ ŞI EXTERNĂ. NUCLEUL UNUI GRAF

Uneori printre vîrfurile unui graf există unele care se bucură de anumite proprietăţi speciale. Pentru caracterizarea acestor submulţimi de vîrfuri, au fost introduse numere speciale, cu o largă aplicabilitate în diferite domenii de cercetare cum sînt: teoria jocurilor, teoria informaţiei şi altele.

Fie graful definit de mulţimea X a vîrfurilor sale şi corespondenţa F, în care să considerăm o submulţime $I \subset X$. Dacă oricare ar fi perechea de vîrfuri din I, nu există un arc care să le lege, se spune că I este o *mulţime interior stabilă*. Cu alte cuvinte mulţimea de vîrfuri $I \subset X$ este *interior stabilă* dacă

$$F\,I \cap I = \varnothing.$$

Să analizăm acum familia $\mathfrak{I}$ a tuturor mulţimilor „interior stabile", ale grafului G.

Fie $|\,I\,|$ numărul elementelor mulţimii I. Vom numi *număr de stabilitate internă* al grafului $G = (X, F)$, $\max\limits_{I \in \mathfrak{I}} |\,I\,| = \alpha(G)$.

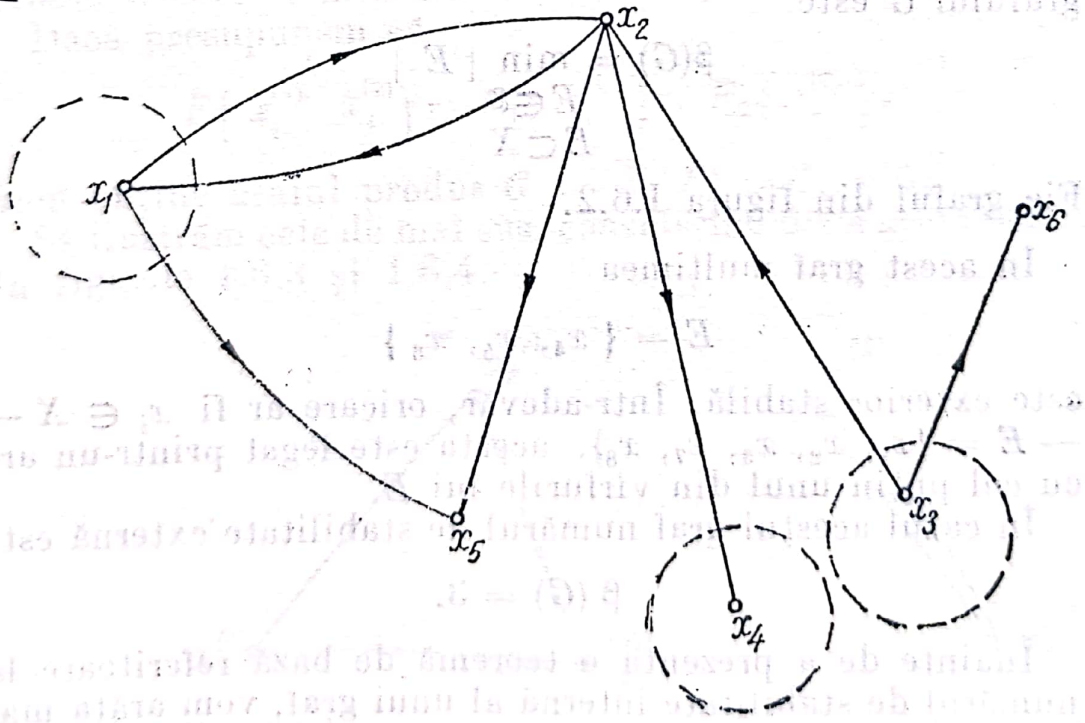


Fig. I.6.1

Pe graful din figura I.6.1. mulțimile

$$I_1 = \{ x_1, x_4, x_3 \},$$
$$I_2 = \{ x_1, x_4, x_6 \},$$
$$I_3 = \{ x_2, x_6 \},$$

sînt mulțimi *interior stabile*.

În adevăr, pentru prima mulțime I_1 avem

$$F_{I_1} = F_{x_1} \cup F_{x_4} \cup F_{x_3} = \{ x_2, x_5 \} \cup \emptyset \cup \{ x_6, x_2 \} = \{ x_2, x_5, x_6 \},$$

$$F_{I_1} \cap I_1 = \{x_1, x_4, x_3 \} \cap \{ x_2, x_5, x_6 \} = \emptyset.$$

Numărul de stabilitate internă pe graful din figura I.6.1 este

$$\alpha(G) = \max_{\substack{I \in \mathcal{I} \\ I \subset X}} |I| = 3.$$

Prin definiție, mulțimea $E \subset X$ este *exterior stabilă*, dacă oricare ar fi $x \in X - E$, avem $F_x \cap E \neq \emptyset$. Cu alte cuvinte, oricare ar fi vîrful x neaparținînd lui E, el are urmași în E. Dacă notăm cu $\mathcal{E}$ familia submulțimilor $E \subset X$ *exterior stabile*, *numărul de stabilitate externă* al grafului G este

$$\beta(G) = \min_{\substack{E \in \mathcal{E} \\ E \subset X}} |E|$$

Fie graful din figura I.6.2.

În acest graf mulțimea

$$E = \{ x_4, x_5, x_6 \}$$

este exterior stabilă. Într-adevăr, oricare ar fi $x_i \in X - E = \{x_1, x_2, x_3, x_7, x_8\}$, acesta este legat printr-un arc cu cel puțin unul din vîrfurile lui E.

În cazul acestui graf numărul de stabilitate externă este

$$\beta(G) = 3.$$

Înainte de a prezenta o teoremă de bază referitoare la numărul de stabilitate internă al unui graf, vom arăta mai întîi ce se înțelege prin produsul a două grafe.

Scanned with OKEN Scanner

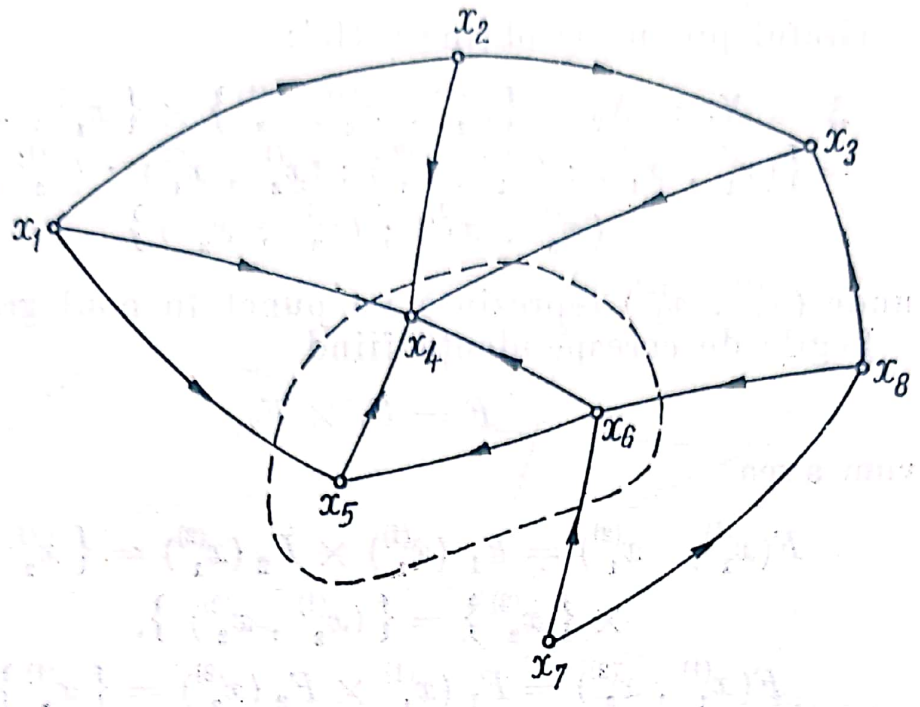


Fig. I.6.2

Fie date două grafe $G_1 = (x_1, F_1)$ și $G_2 = (X_2\ F_2)$. Să cercetăm graful construit cu elementele mulțimii $X = X_1 \times X_2$, unde mulțimea X, după cum se vede, reprezintă produsul cartezian al celor două mulțimi.

Dacă presupunem că

$$F\left(x_i^{(1)}, x_j^{(2)}\right) = F_1\left(x_i^{(1)}\right) \times F_2\left(x_j^{(2)}\right),$$

vom obține graful produs $G = (X, F) = G_1 \times G_2$.

Să ilustrăm cele de mai sus, considerînd două grafe redate în figurile I.6.3 și I.6.4.

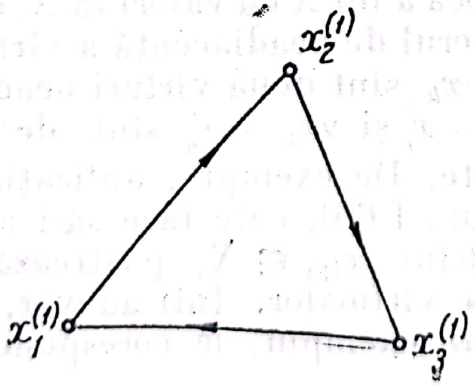


Fig. I.6.3

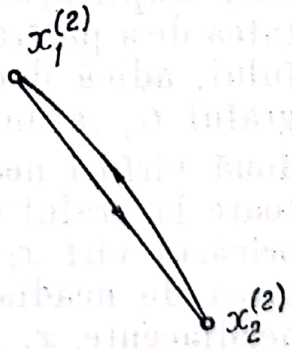


Fig. I.6.4

Graful produs se obține astfel:

$$X = X_1 \times X_2 = \{ x_1^{(1)}, x_2^{(1)}, x_3^{(1)} \} \times \{ x_1^{(2)}, x_2^{(2)} \} =$$
$$= \{ (x_1^{(1)}, x_1^{(2)}) ; (x_1^{(1)}, x_2^{(2)}) ; (x_2^{(1)}, x_1^{(2)}) ; (x_2^{(1)}, x_2^{(2)}) ;$$
$$(x_3^{(1)}, x_1^{(2)}) ; (x_3^{(1)}, x_2^{(2)}) \}$$

unde $(x_i^{(1)}, x_j^{(2)})$ reprezintă un punct în noul graf.

Legile de corespondență fiind

$$F = F_1 \times F_2,$$

vom avea

$$F(x_1^{(1)}, x_1^{(2)}) = F_1(x_1^{(1)}) \times F_2(x_1^{(2)}) = \{ x_2^{(1)} \} \times$$
$$\times \{ x_2^{(2)} \} = \{ (x_2^{(1)}, x_2^{(2)}) \},$$
$$F(x_1^{(1)}, x_2^{(2)}) = F_1(x_1^{(1)} \times F_2(x_2^{(2)}) = \{ x_2^{(1)} \} \times$$
$$\times \{ x_1^{(2)} \} = \{ (x_2^{(1)} \times x_1^{(2)}) \},$$
$$F(x_2^{(1)}, x_1^{(2)}) = F_1(x_2^{(1)}) \times F_2(x_1^{(2)}) = \{ x_3^{(1)} \} \times$$
$$\times \{ x_2^{(2)} \} = \{ (x_3^{(1)}, x_2^{(2)}) \},$$

$$F(x_2^{(1)}, x_2^{(2)}) = F_1(x_2^{(1)}) \times F_2(x_2^{(2)}) = \{x_3^{(1)}\} \times \{x_1^{(2)}\} = \{(x_3^{(1)}, x_1^{(2)})\},$$

$$F(x_3^{(1)}, x_1^{(2)}) = F_1(x_3^{(1)}) \times F_2(x_1^{(2)}) = \{x_1^{(1)}\} \times \{x_2^{(2)}\} = \{(x_1^{(1)}, x_2^{(2)})\},$$

$$F(x_3^{(1)}, x_2^{(2)}) = F_1(x_3^{(1)}) \times F_2(x_2^{(2)}) = \{x_1^{(1)}\} \times \{x_1^{(2)}\} = \{(x_1^{(1)}, x_1^{(2)})\}.$$

Graful produs este redat în figura I.6.5.

Fie dat graful $G = (X, F)$. În cele ce urmează vom nota cu ν o aplicație biunivocă a lui X cu valori în X, cu proprietatea de a păstra caracterul de neadiacență a vîrfurilor grafului, adică dacă x_i și x_k sînt două vîrfuri neadiacente în graful G, atunci $\nu x_i = x_i'$ și $\nu x_k = x_k'$ sînt de asemenea două vîrfuri neadiacente. De exemplu, aplicația ν referitoare la graful din figura I.6.6, care face să-i corespundă oricărui vîrf $x_i \in X$ vîrful $x_{i+1} \in X$, păstrează proprietatea de neadiacență a vîrfurilor. Într-adevăr, vîrfurilor neadiacente x_1 și x_3, de exemplu, le corespund

$$\nu x_1 = x_2 \quad \text{și} \quad \nu x_3 = x_4,$$

Scanned with OKEN Scanner

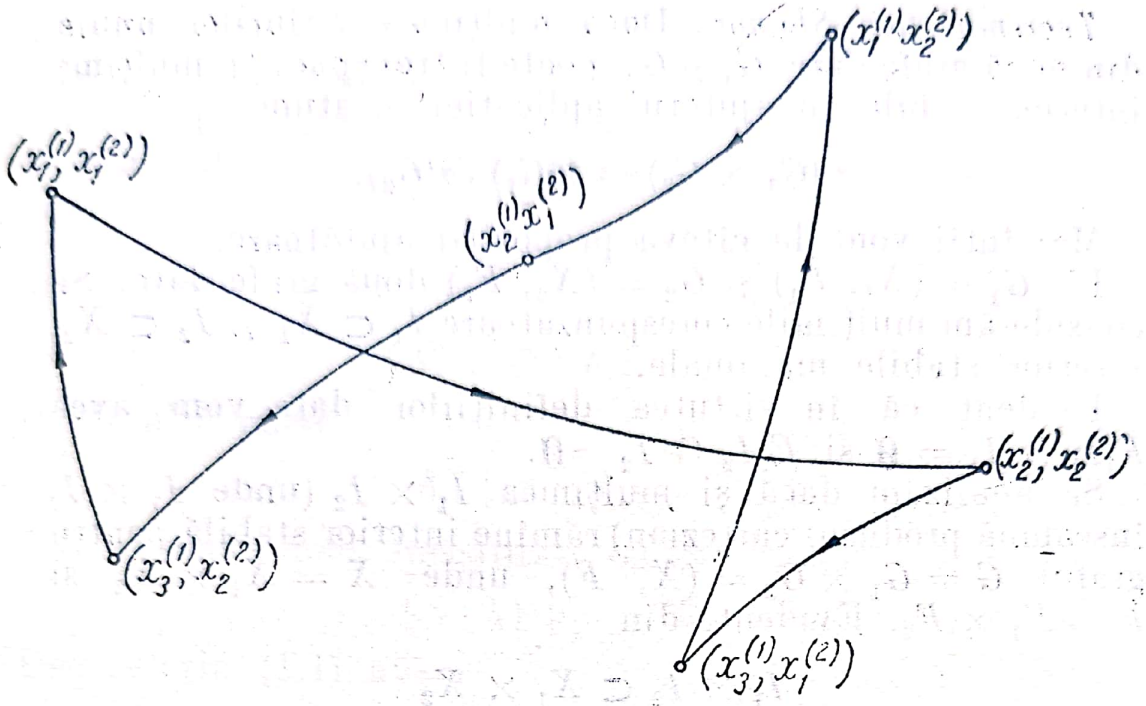


Fig 1.6.5

x_2 și x_4 fiind de asemenea neadiacente. Dacă se consideră mulțimea $X = \{x_1, x_2, x_3, x_4\}$, atunci $\nu(X) = \{x_2, x_3, x_4, x_5, x_1\}$.

Cu ajutorul noțiunilor definite, dăm mai departe o teoremă importantă referitoare la numărul de stabilitate internă.

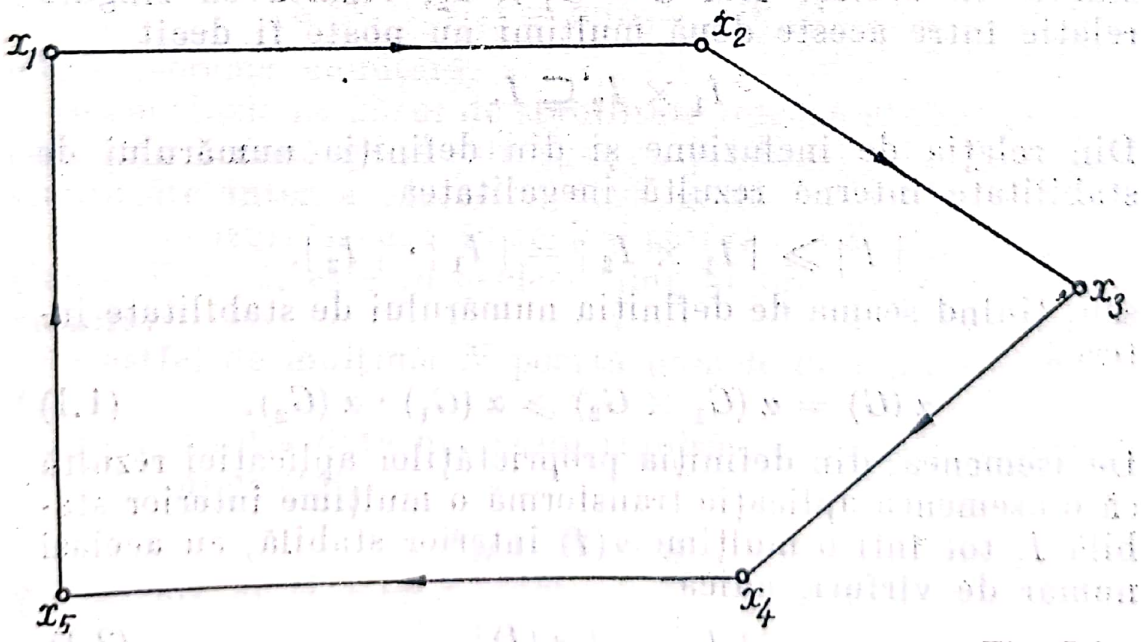


Fig. I.6.6

Teorema lui Shanon. Dacă mulțimea vîrfurilor unuia din două grafe date, G_1 și G_2, poate fi transpusă în mulțime interior stabilă cu ajutorul aplicației ν, atunci

$$\alpha(G_1 \times G_2) = \alpha(G_1) \cdot \alpha(G_2).$$

Mai întîi vom da cîteva propoziții ajutătoare.

Fie $G_1 = (X_1, F_1)$ și $G_2 = (X_2, F_2)$ două grafe date. Să considerăm mulțimile corespunzătoare $I_1 \subset X_1$ și $I_2 \subset X_2$, interior stabile maximale.

Evident că în virtutea definițiilor date vom avea $F_1 I_1 \cap I_1 = \emptyset$ și $F_2 I_2 \cap I_2 = \emptyset$.

Să analizăm dacă și mulțimea $I_1 \times I_2$ (unde $I_1 \times I_2$ înseamnă produsul cartezian) rămîne interior stabilă pentru graful $G = G_1 \times G_2 = (X, F)$, unde $X = X_1 \times X_2$ și $F = F_1 \times F_2$. Evident, din

$$I_1 \times I_2 \subset X_1 \times X_2$$

$$(I_1 \times I_2) \cap F(I_1 \times I_2) = (I_1 \times I_2) \cap (F_1 I_1 \times F_2 I_2) = \\ = (I_1 \cap F_1 I_1) \times (I_2 \cap F_2 I_2) = \emptyset \times \emptyset = \emptyset.$$

Ultima egalitate afirmă că mulțimea $(I_1 \times I_2)$ este de asemenea interior stabilă în graful produs $G_1 \times G_2$.

Să notăm cu I mulțimea maximală, interior stabilă din graful $G_1 \times G_2 = G$. Cum mulțimea $I_1 \times I_2$ este interior stabilă în același graf $G = G_1 \times G_2$, rezultă că singura relație între aceste două mulțimi nu poate fi decît

$$I_1 \times I_2 \subseteq I.$$

Din relația de incluziune și din definiția numărului de stabilitate internă rezultă inegalitatea

$$|I| \geqslant |I_1 \times I_2| = |I_1| \cdot |I_2|,$$

sau, ținînd seama de definiția numărului de stabilitate internă,

$$\alpha(G) = \alpha(G_1 \times G_2) \geqslant \alpha(G_1) \cdot \alpha(G_2). \qquad (1.I)$$

De asemenea, din definiția proprietăților aplicației rezultă că o asemenea aplicație transformă o mulțime interior stabilă I, tot într-o mulțime $\nu(I)$ interior stabilă, cu același număr de vîrfuri, adică

$$|I| = |\nu(I)|. \qquad (2.I)$$

Scanned with OKEN Scanner

Să considerăm mulțimea X a vîrfurilor grafului G și aplicația ν definită pe X.

Dacă $\nu(X)$ este o mulțime interior stabilă, vom arăta că numărul $|\nu(X)|$ este același cu numărul de stabilitate internă a grafului G.

Fie I mulțimea interior stabilă din G cu număr maxim de elemente. În acest caz

$$|\nu(X)| \leqslant |I| = \alpha(G). \tag{3.I}$$

Pe de altă parte

$$|\nu(X)| \geqslant |I| = \alpha(G).$$

Din ultimele două inegalități rezultă egalitatea

$$|\nu(X)| = \alpha(G).$$

Din relația (1.I) avem

$$\alpha(G_1 \times G_2) \geqslant \alpha(G_1) \cdot \alpha(G_2). \tag{4.I}$$

În baza relației (3.I) avem

$$\alpha(G_1 \times G_2) \leqslant \alpha(G_1) \cdot \alpha(G_2). \tag{5.I}$$

Deoarece produsul cartezian transformă o mulțime interior stabilă tot într-o mulțime interior stabilă, din compararea relațiilor (4.I) și (5.I) obținem

$$\alpha(G_1 \times G_2) = \alpha(G_1) \cdot \alpha(G_2),$$

adică teorema enunțată.

În concluzie numărul de stabilitate internă al produsului cartezian a două grafe este egal cu produsul numerelor de stabilitate internă ale celor două grafe.

Există cazuri în care, în același graf $G = (X, F)$, submulțimea $N \subset X$, este în același timp și interior și exterior stabilă.

O astfel de mulțime N poartă numele de *nucleu al grafului*.

Din definiția dată nucleului rezultă că acesta are următoarele proprietăți:

$$F_N \cap N = \emptyset$$

și, oricare ar fi $x \in X - N$,

$$F_x \cap N \neq \emptyset.$$

Scanned with OKEN Scanner

Proprietatea 1. Dacă graful $G = (X, F)$ are nucleul N, între numerele lui de stabilitate internă şi externă există relaţia

$$\alpha(G) = \max_{I \in \mathcal{I}} |I| \geqslant |N_x| \geqslant \min_{E \in \mathcal{E}} |E| = \beta(G).$$

Într-adevăr, prin definiţie, numărul de stabilitate internă

$$\alpha(G) \max_{I \in \mathcal{I}} |I|.$$

Mulţimea N, fiind o mulţime interior stabilă, adică $N_x \in \mathcal{I}$, avem

$$|N| \leqslant \max_{I \in \mathcal{I}} |I| = \alpha(G). \qquad (6.I)$$

Pe de altă parte, N fiind exterior stabilă, adică $N \in \mathcal{E}$ avem şi

$$|N| \geqslant \min_{N \in \mathcal{E}} |E| = \beta(G). \qquad (7.I)$$

Comparînd (6.I) cu (7.I) obţinem relaţia

$$\alpha(G) = \max_{I \in \mathcal{I}} |I| \geqslant |N| \geqslant \min_{E \in \mathcal{E}} |E| = \beta(G).$$

În graful din figura I.6.7 mulţimea $N = \{x_3, x_7, x_5\}$ formează un nucleu.

Evident că nu orice graf are nucleu.

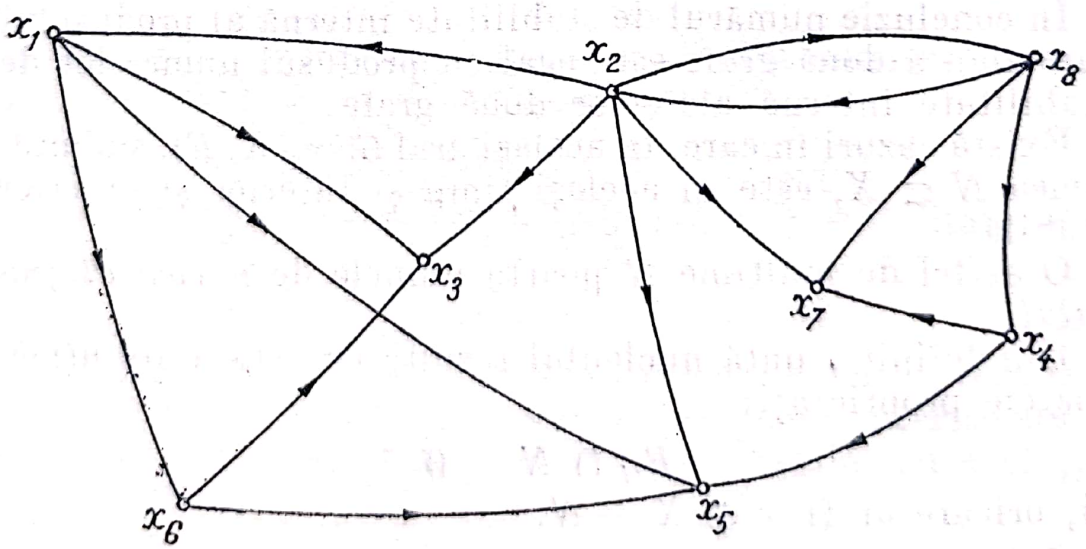


Fig. I.6.7

Scanned with OKEN Scanner

Teoremă. Dacă N este nucleul grafului $G = (X, F)$, atunci mulțimea N este maximală în familia $\mathcal{J}$, adică dacă $I \in \mathcal{J}$ și dacă $I \supset N$, atunci $I = N$. Cu alte cuvinte, nucleul unui graf N nu este submulțime proprie pentru mulțimile interior stabile ale grafului.

Într-adevăr, să presupunem că există o mulțime $I \in \mathcal{J}$, astfel că $N \subset I$. Din incluziunea $N \subset I$ rezultă că există cel puțin un vîrf $x_i \in I$, dar $x_i \notin N$. Deoarece N este și exterior stabilă, rezultă $F_{x_i} \cap N \neq \emptyset$, de unde $F_{x_i} \cap I \neq \emptyset$, ceea ce contrazice ipoteza că mulțimea I ar fi interior stabilă, de unde rezultă teorema enunțată.

Noțiunile de *stabilitate internă* și *stabilitate externă* își găsesc aplicații interesante în teoria informației.

După cum se știe, în toate domeniile de activitate, astăzi mai mult ca oricînd, se cere să se transmită o cantitate cît mai mare de informații și, dacă e posibil, într-un timp cît mai scurt.

Teoria informației este capabilă, în prezent, prin mijloacele de care dispune, să dea răspunsul la asemenea probleme. În procesul de transmitere apar în mod inerent perturbații care pot să ducă la erori și să denatureze informația transmisă. Cu ajutorul teoriei grafelor, mai precis cu ajutorul numărului de stabilitate internă, poate fi determinat numărul maxim de semnale transmise, astfel încît la recepție să nu apară confuzii.

§ 7. FUNCȚIA LUI GRUNDY ȘI DETERMINAREA NUCLEELOR UNUI GRAF

Dată fiind importanța nucleelor unui graf în aplicațiile legate de teoria jocurilor, precum și în alte domenii, vom prezenta în acest paragraf, o metodă de determinare a nucleelor unui graf cu ajutorul funcției lui Grundy.

Fie graful $G = (X, F)$, finit sau infinit. O funcție g definită pe mulțimea X, cu valori în mulțimea N^* ($N^* = N \cup \{0\}$) se numește *funcția lui Grundy* dacă:

— $g(x) = 0$, pentru orice $x \in X$, cu $F_x = \emptyset$,

— $g(x) = \inf \cdot N^* \notin g(F_x)$.

Scanned with OKEN Scanner

Pentru determinarea valorilor funcției $g(x)$, în aplicații se procedează astfel. Considerăm submulțimea $X_0 \subset X$, pentru care $FX_0 = \emptyset$; în acest caz, în baza definiției, pentru orice $x_i^{(0)} \in X_0$ avem

$$g(x_i^{(0)}) = 0.$$

Considerăm apoi submulțimea $X_1 \subset X$, care are urmași numai în X_0; pentru orice vîrf $x^{(1)} \in X_1$ avem

$$g(x^{(1)}) = 1.$$

În continuare se ia submulțimea $X_2 \subset X$, care are urmași numai în X_0 sau X_1, sau în X_0 și X_1. În acest caz apar două situații distincte:

a) Un vîrf $x_i^{(2)}$ are urmași și în submulțimea X_1 și în submulțimea X_0; atunci se ia

$$g(x_i^{(2)}) = 2.$$

b) Vîrful $x_i^{(2)}$ are urmași numai în submulțimea X_1; atunci se ia

$$g(x_i^{(2)}) = 0.$$

Procedînd din aproape în aproape în modul descris, se ajunge la o submulțime $X_k \subset X$, pentru care

$$FX_k \subset X_0 \cup X_1 \cup X_2 \cup \ldots X_{k-1}.$$

De aici, pentru orice vîrf $x_i^{(k)} \in X_k$, $g(x_i^{(k)}) = \inf N^* \in \notin \{g(F_{x_i}{}^{(k)})\}$

Să aplicăm cele prezentate considerînd graful din figura I.7.1:

$$X_0 = \{x_9, x_{10}\},\ F_{x_9} = \emptyset \text{ și } F_{x_{10}} = \emptyset,$$

deci

$$g(x_9) = 0 \text{ și } g(x_{10}) = 0.$$

$$X_1 = \{x_6,\},\ F_{x_6} = x_{10} \in X_0,$$

deci

$$g(x_6) = 1.$$

$$X_2 = \{x_7\},\ F_{x7} = \{x_6, x_9, x_{10}\} = \{x_6\} \cup \{x_9, x_{10}\} = X_1 \cup X_0,$$

Scanned with OKEN Scanner

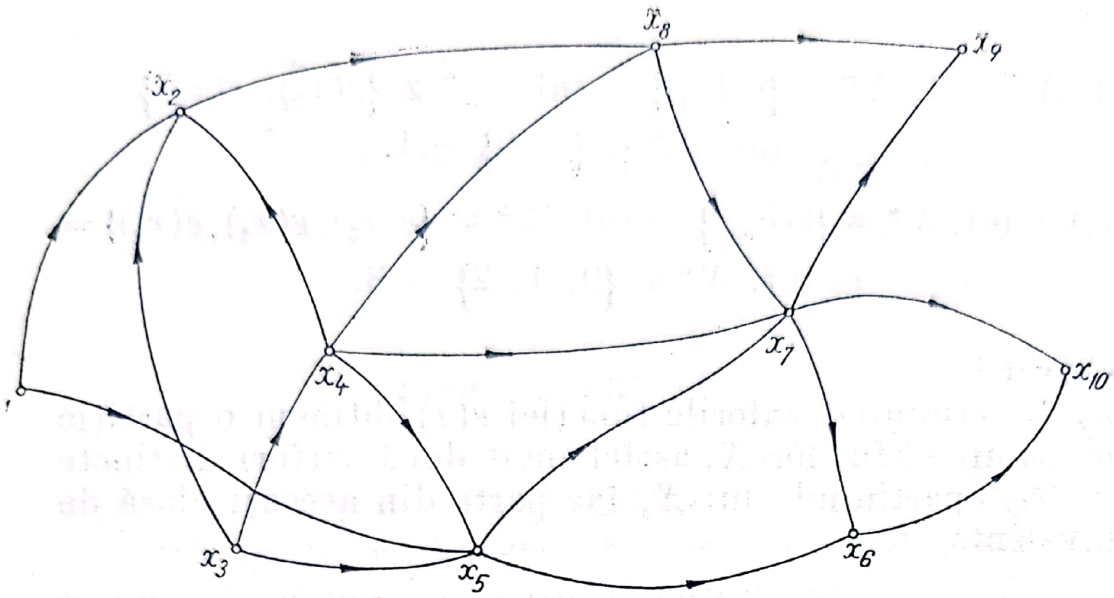


Fig. I.7.1

deci

$$g(x_7) = \text{inf. } N^* \notin \{g(F_{x_7})\} = \text{inf. } N^* \notin \{0, 1\} = 2.$$

$$X_3 = \{x_5, x_8\},\ F_{x_5} = \{x_6, x_7\} \subseteq X_1 \cup X_2$$
$$\text{și } F_{x_8} = \{x_7, x_9\} \subset X_2 \cup X_0,$$

deci

$$g(x_5) = \text{inf. } N^* \notin \{g(F_{x_5})\} = \text{inf. } N^* \notin \{g(x_6), g(x_7)\} =$$
$$= \text{inf. } N^* \notin \{1, 2\} = 0,$$

$$g(x_8) = \text{inf. } N^* \notin \{g(F_{x_8})\} = \text{inf. } N^* \notin \{g(x_7), g(x_9)\} =$$
$$= \text{inf. } N^* \notin \{2, 0\} = 1.$$

$$X_4 = \{x_2, x_4\},\ F_{x_2} = \{x_8\} \subset X_3 \text{ și } F_{x_4} = \{x_5, x_7, x_8\},$$

deci

$$g(x_2) = \text{inf. } N^* \notin \{g(F_{x_2})\} = \text{inf. } N^* \notin \{1\} = 0,$$

$$g(x_4) = \text{inf. } N^* \notin \{g(F_{x_4})\} = \text{inf. } N^* \notin \{g(x_5), g(x_7), g(x_8)\}$$
$$= \text{inf. } N^* \notin \{0, 2, 1\} = 3.$$

$$X_5 = \{x_1, x_3\},\ F_{x_1} = \{x_2, x_5\} \text{ și } F_{x_3} = \{x_5, x_4, x_2\},$$

Scanned with OKEN Scanner

deci

$$g(x_1) = \text{inf. } N^* \notin \{g(F_{x_1})\} = \text{inf. } N_x^* \notin \{g(x_2),\ g(x_5)\} =$$
$$= \text{inf. } N^* \notin \{2,\ 0\} = 1.$$

$$g(x_3) = \text{inf. } N^* \notin \{g(F_{x_3})\} = \text{inf. } N^* \notin \{g(x_5), g(x_4), g(x_2)\} =$$
$$= \text{inf. } N^* \notin \{0,\ 1,\ 2\} = 3.$$

Observații:

a) Determinînd valorile funcției $g(x)$ obținem o partiție a mulțimii vîrfurilor X, astfel încît două vîrfuri distincte x_i și x_k, aparținînd lui X, fac parte din aceeași clasă de echivalență dacă

$$g(x_i) = g(x_k).$$

b) Mulțimea vîrfurilor

$$N = \{x_{i_1},\ x_{i_2},\ldots,\ x_{i_k},\ldots,\ x_{i_p}\},$$

pentru care

$$g(x_{i_k}) = 0,\ k = 1,\ 2,\ldots,\ p,$$

formează *nucleul grafului*. Într-adevăr, oricare ar fi vîrful x_{i_k}, avem

$$F_{x_{i_k}} \cap N = \emptyset,$$

deoarece în caz contrar ar trebui să avem

$$g(x_{i_k}) > 0,$$

ceea ce ar contrazice ipoteza făcută.

Dacă x este un vîrf ce nu aparține mulțimii N, adică

$$x \in X - N,$$

înseamnă că, $g(x) > 0$; aceasta însă antrenează

$$F_x \cap N \neq \emptyset.$$

Se constată că mulțimea N satisface cele două condiții caracteristice nucleului unui graf.

În cazul grafului din figura I.7.1 mulțimea $N = \{x_2, x_5, x_9, x_{10}\}$ formează un nucleu.

Scanned with OKEN Scanner

CAPITOLUL

2

DRUMURI ÎNTR-UN GRAF

§ 1. INTRODUCERE

Noțiunea de drum este o noțiune deosebit de importantă în teoria grafelor, deoarece ea este strîns legată de rezolvarea multor probleme cu caracter aplicativ, printre care cele referitoare la găsirea timpului maxim de realizare a unui proiect, stabilirea ordinei de lansare în producție a unor repere, determinarea lungimii minime a cablului care leagă mai multe orașe între ele (în sistemul de telecomunicații), probleme de aprovizionare, precum și multe altele de acest gen.

Reamintim că drumul este o succesiune de arce, în care extremitatea finală a unuia coincide cu extremitatea inițială a următorului și că lungimea sa este dată de numărul de aree care intră în componența lui, sau de suma lungimilor arcelor care-l compun.

Deseori, în problemele practice, apare necesitatea de a alege, din mulțimea de drumuri posibile dintre două vîrfuri ale grafului, pe acela care are lungime minimă sau maximă în funcție de caracterul concret al problemei.

Astfel, dacă vîrfurile grafului reprezintă localități care urmează să fie legate cu cablu de transmitere a informației, se impune găsirea drumului de lungime minimă (cablu de lungime minimă) care să lege toate orașele între ele. Dacă se pune problema construirii unei căi de comunicație (șosea, cale ferată) între două centre principale, din mulțimea de posibilități se alege aceea care să cuprindă anumite localități intermediare, dar în așa fel, încît cheltuielile totale afectate construirii traseului să fie minime.

În alte situații, cum sînt cele legate de execuția unui proiect — unde vîrfurile grafului reprezintă sfîrșitul unor

Scanned with OKEN Scanner

activităţi şi începutul altora, iar arcele reprezintă ordinea de efectuare a acestora — se cere, dimpotrivă, drumul de lungime maximă dintre vîrful care reprezintă prima operaţie şi vîrful care reprezintă ultima operaţie, respectiv terminarea lucrării.

Pentru a da răspuns unor asemenea probleme, vom prezenta în cele ce urmează unele metode de găsire a drumului de lungime minimă sau maximă, între două vîrfuri date ale unui graf.

§ 2. DRUMUL DE LUNGIME MINIMĂ ÎNTRE DOUĂ VÎRFURI ALE UNUI GRAF

Fie dat un graf $G = (X, F)$, unde fiecărui arc a_{ij} îi asociem o valoare $C_{ij} \geqslant 0$.

Ne propunem să găsim drumul dintre două vîrfuri oarecare x_k şi x_q ale grafului, astfel ca ΣC_{ij} corespunzătoare arcelor drumului să fie minimă sau maximă.

Numărul C_{ij} poate însemna fie costul transportului de la x_i la x_j, fie cheltuielile aferente construcţiei drumului dintre aceste două puncte, fie încă, timpul necesar mersului de la x_i la x_j.

La baza rezolvării unor asemenea probleme stă teorema de optimalitate a lui Bellman, care afirmă că drumul optimal (maxim sau minim) nu poate conţine decît drumuri parţiale optimale (maxime sau minime).

Să considerăm nouă centre industriale notate respectiv cu x_1, x_2,..., x_9 şi să presupunem cunoscute duratele medii pentru parcurgerea traseelor între diversele centre acolo unde traficul este posibil. Evident că există multiple posibilităţi de a parcurge drumul dintre x_1 şi x_9. Din mulţimea de drumuri posibile care leagă pe x_1 de x_9, să se aleagă cel de lungime minimă.

Graful din figura II.2.1 reprezintă localităţile, traseele posibile, precum şi duratele medii de parcurgere. Să admitem că traseele se parcurg într-un singur sens. Drumul de lungime minimă (durată minimă) dintre vîrfurile x_1 şi x_8 îl vom calcula cu ajutorul teoremei lui Bellman. Pentru aceasta,

Scanned with OKEN Scanner

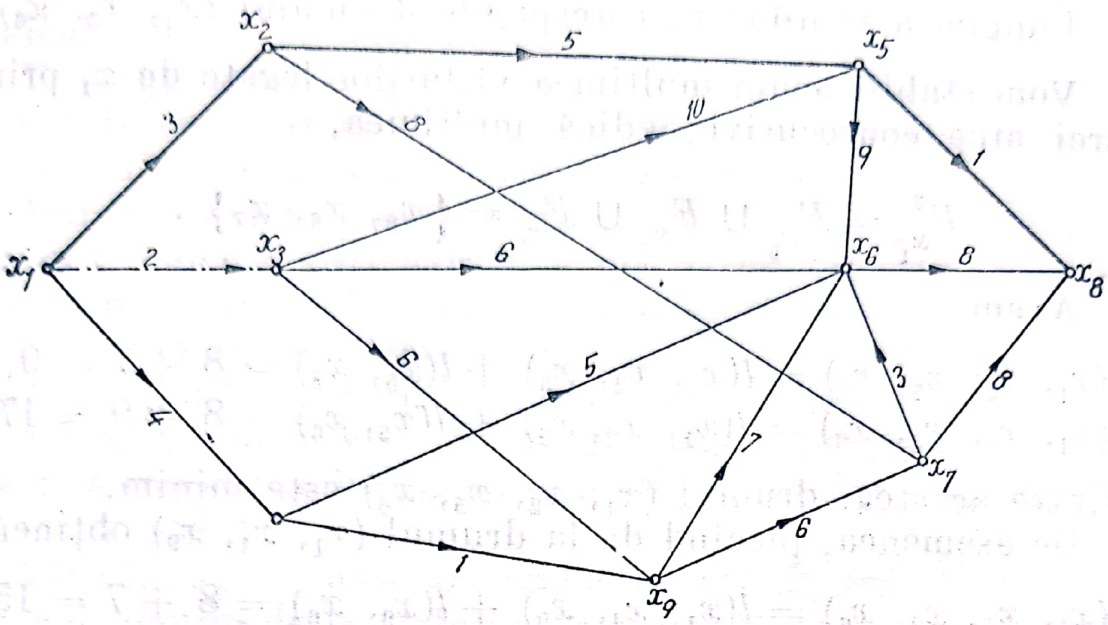


Fig. II.2.1

vom calcula mai întîi lungimea celui mai scurt drum care conduce la fiecare vîrf al grafului, plecînd din x_1.

Observăm că

$$F^2_{x_1} = F_{x_2} \cup F_{x_3} \cup F_{x_4} = \{x_5,\ x_7,\ x_9\ x_6\}\ .$$

Aceste vîrfuri sînt legate de x_1 prin două arce. Lungimile drumurilor posibile de la x_1 la vîrfurile mulțimii $F^2_{x_1}$ sînt

$$l(x_1, x_5) = \begin{cases} 2 + 10 = 12, \text{ prin } x_3; \\ 3 + 5 = 8, \text{ prin } x_2. \end{cases}$$

În baza teoremei, vom reține drumul care are lungimea cea mai mică, adică drumul parțial (x_1, x_2, x_5) a cărui lungime, în exemplul nostru, este 8.

$$l(x_1, x_7) = 3 + 8 = 11, \text{ prin } x_2.$$

Evident că fiind singurul drum de la x_1, la x_7, îl reținem:

$$l(x_1, x_9) = \begin{cases} 4 + 1 = 5, \text{ prin } x_4; \\ 2 + 6 = 8, \text{ prin } x_3. \end{cases}$$

Reținem drumul (x_1, x_4, x_9) care are lungimea cea mai mică, de 5 unități.

$$l(x_1, x_6) = \begin{cases} 2 + 6 = 8, \text{ prin } x_3; \\ 4 + 5 = 9, \text{ prin } x_4. \end{cases}$$

Lungimea minimă 8, corespunde drumului (x_1, x_3, x_6).

Vom stabili acum mulțimea vîrfurilor legate de x_1 prin trei arce consecutive, adică mulțimea,

$$F^3_{x_1} = F_{x_5} \cup F_{x_7} \cup F_{x_9} = \{x_8, x_6, x_7\}.$$

Avem

$$l(x_1, x_2, x_5, x_8) = l(x_1, x_2, x_5) + l(x_5, x_8) = 8 + 1 = 9,$$
$$l(x_1, x_2, x_5, x_6) = l(x_1, x_2, x_5) + l(x_5, x_6) = 8 + 9 = 17;$$

dintre acestea, drumul (x_1, x_2, x_5, x_8) este minim.

De asemenea, plecînd de la drumul (x_1, x_4, x_9) obținem

$$l(x_1, x_4, x_9, x_6) = l(x_1, x_4, x_9) + l(x_9, x_6) = 8 + 7 = 15,$$
$$l(x_1, x_4, x_9, x_7) = l(x_1, x_4, x_9) + l(x_9, x_7) = 8 + 6 = 14.$$

Ultimul drum este cel minim. În sfîrșit avem

$$l(x_1, x_3, x_6, x_8) = l(x_1, x_3, x_6) + l(x_6, x_8) = 8 + 8 = 16,$$

drum unic și ca atare îl reținem. În final, luăm în considerație drumurile care duc la x_8 și care sînt

$$l(x_1, x_2, x_5, x_8) = 9,$$
$$l(x_1, x_4, x_9, x_7, x_8) = l(x_1, x_4, x_9, x_7) + l(x_7, x_8) = 14 + 8 = 22,$$
$$l(x_1, x_3, x_6, x_8) = 16.$$

Dintre cele trei drumuri posibile, cel de lungime minimă este (x_1, x_2, x_5, x_8). Se constată ușor că acesta este compus din drumuri parțiale minime.

§ 3. ALGORITME PENTRU GĂSIREA DRUMULUI DE LUNGIME MINIMĂ DINTRE DOUĂ VÎRFURI ALE UNUI GRAF

1°. Algoritmul lui Ford. Aplicarea teoremei lui Bellman ne-a condus la găsirea unor drumuri de lungime minimă, între x_1 și fiecare vîrf al grafului. Însă pentru un graf cu

Scanned with OKEN Scanner

un număr mare de vîrfuri o astfel de metodă conduce la calcule lungi și susceptibile de erori.

Evident că din moment ce s-a determinat drumul cel mai scurt dintre x_0 și x_1, x_2,..., x_k este inutil să mai calculăm distanța pînă la x_{k+1}, pornind din x_0.

Matematicianul Ford, printre numeroasele algoritme care permit calcule sistematice, a dat și un algoritm extrem de simplu, cunoscut în literatura de specialitate sub numele de *algoritmul lui Ford*.

Algoritmul lui Ford propune într-o *primă etapă* să se asocieze fiecărui vîrf x_i al grafului numerele $\lambda_i \geqslant 0$, care să reprezinte lungimea unui drum arbitrar de la vîrful x_0 la vîrful x_i.

Vîrfului x_0 i se asociază numărul $\lambda_0 = 0$, deoarece lungimea drumului de la vîrful x_0 la el însuși este evident egală cu zero.

Dacă numerele λ_i ar fi alese de așa manieră încît ele să reprezinte lungimea minimă a drumului de la vîrful x_0 la vîrful x_i, numărul λ_n, asociat vîrfului x_n, ar fi tocmai lungimea drumului minim dintre vîrfurile x_0 și x_n.

În grafele cu multe vîrfuri această operație este greoaie și uneori practic imposibilă. De aceea, în prima etapă numerele λ_i se iau arbitrar și apoi se trece la *etapa a doua*. În această etapă se calculează diferențele $\lambda_j - \lambda_i$, începînd cu λ_n, diferențe ce se compară cu lungimile $l(x_i, x_j)$. Pentru fiecare arc poate exista una din următoarele trei situații:

$$\begin{aligned} &\text{(a)}\ \lambda_j - \lambda_i < l(x_i, x_j), \\ &\text{(b)}\ \lambda_j - \lambda_i = l(x_i, x_j), \\ &\text{(c)}\ \lambda_j - \lambda_i > l(x_i, x_j). \end{aligned}$$

Din inegalitatea (a) rezultă

$$\lambda_j < \lambda_i + l(x_i, x_j).$$

În figura II.3.1 se observă că drumul de la x_0 la x_j, de lungime λ_j, este mai scurt decît drumul care trece prin x_i.

Egalitatea (b) exprimă faptul că cele două drumuri, de la x_0 la x_j, au aceeași lungime, deoarece

$$\lambda_j = \lambda_i + l(x_i, x_j).$$

Scanned with OKEN Scanner

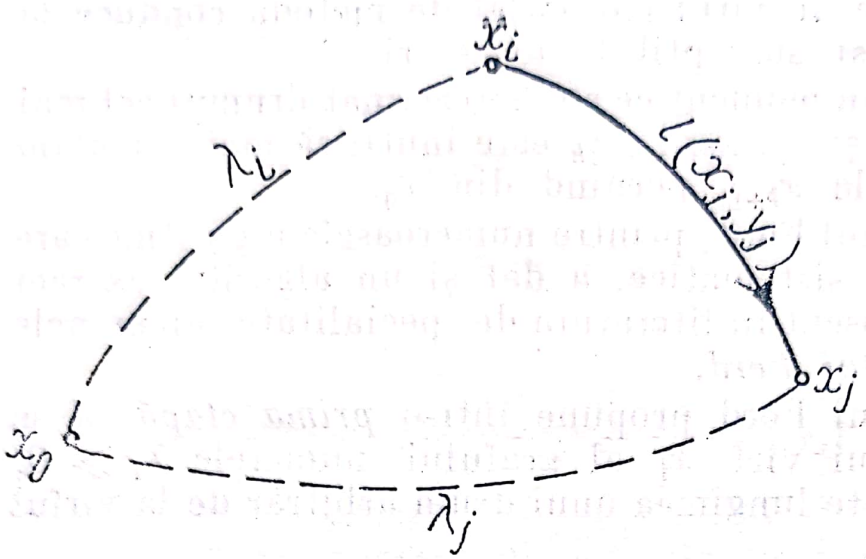


Fig. II.3.1

În cazul inegalității (c) avem

$$\lambda_j > \lambda_i + l(x_i, x_j),$$

ceea ce arată că drumul de la x_0 la x_j, prin x_i, are lungimea mai mică decît drumul de la x_0 direct la x_j.

Întrucît se urmărește determinarea drumului de lungime minimă între x_0 și x_j, vom lua în considerație drumul care trece prin x_i, a cărui lungime este,

$$\lambda'_j = \lambda_i + l(x_i, x_j) < \lambda_j.$$

În consecință, după efectuarea tuturor comparațiilor posibile, vom trece la modificarea numai a acelor λ_j pentru care este valabilă inegalitatea (c), celelalte rămînînd neschimbate.

Cu noile valori λ_j se repetă operația de comparare, pînă cînd, după un număr finit de ori, k, toate diferențele verifică inegalitatea

$$\text{(d)} \quad \lambda_j^{(k)} - \lambda_i^{(k)} \leqslant l(x_i, x_j).$$

Din acest moment se trece la alegerea drumului minim, luînd în considerație, din șirul de diferențe (d), numai pe acelea pentru care este valabilă egalitatea

$$\lambda_j^{(k)} - \lambda_i^{(k)} = l(x_i, x_j).$$

Scanned with OKEN Scanner

Să admitem că acestea sînt:

$$\lambda_n^{(k)} - \lambda_{i_1}^{(k)} = l(x_{i_1}, x_n),$$
$$\lambda_{i_1}^{(k)} - \lambda_{i_2}^{(k)} = l(x_{i_2}, x_{i_1}),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\lambda_{i_{p-1}}^{(k)} - \lambda_{i_p}^{(k)} = l(x_{i_p}, x_{i_{p-1}}),$$
$$\lambda_{i_p}^{(k)} - \lambda_0^{(k)} = l(x_0, x_{i_p}).$$

Dintre numerele $\lambda_j^{(k)}$ vom alege pe acelea ai căror indici formează un șir monoton descrescător. În cazul în care în șirul $\lambda_j^{(k)}$ o valoare se repetă de q ori, înseamnă că în graf există q drumuri cu aceeași lungime minimă.

Exemplu. Să admitem că între localitățile x_0 și x_{11} se pune problema construirii unei șosele care are posibilitatea să treacă prin localitățile x_1, x_2,..., x_{10}.

Deoarece cheltuielile necesare construirii porțiunilor de șosea dintre diversele localități diferă în funcție de mulți factori (teren, utilaje, apropiere de centru, de aprovizionare, etc.); se cere să se determine traseul șoselei astfel încît costul total al lucrării să fie minim.

În graful din figura II.3.2 am reprezentat prin puncte localitățile, iar prin arce porțiunile posibile de șosea, iar pe fiecare arc este trecut costul afectat lucrării respective.

Găsirea traseului de cost total minim se reduce la determinarea drumului de lungime minimă din acest graf. Să aplicăm acum algoritmul lui Ford pentru determinarea drumului minim de la x_0 la x_{11}.

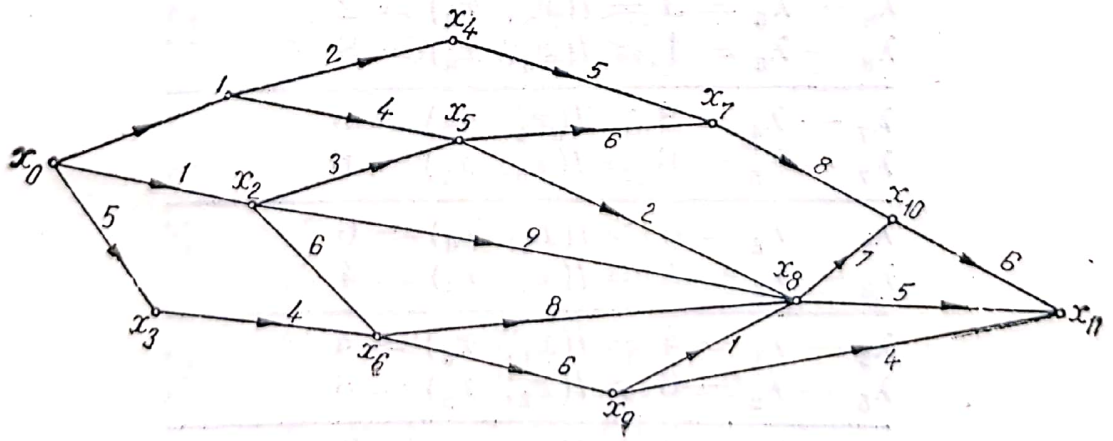


Fig. II.3.2

Etapa I. Atașăm vîrfurilor grafului numerele

$\lambda_0 = 0$
$\lambda_1 = 3$
$\lambda_2 = 1$
$\lambda_3 = 5,$
$\lambda_4 = l(x_0,\ x_1,\ x_4) = 3 + 2 = 5,$
$\lambda_5 = l(x_0,\ x_1,\ x_5) = 3 + 4 = 7,$
$\lambda_6 = l(x_0,\ x_3,\ x_6) = 5 + 4 = 9,$
$\lambda_7 = l(x_0,\ x_1,\ x_5,\ x_7) = 3 + 4 + 6 = 13,$
$\lambda_8 = l(x_0,\ x_2,\ x_8) = 1 + 9 = 10,$
$\lambda_9 = l(x_0,\ x_3,\ x_6,\ x_9) = 5 + 4 + 6 = 15,$
$\lambda_{10} = l(x_0,\ x_1,\ x_5,\ x_7,\ x_{10}) = 3 + 4 + 6 + 8 = 21,$
$\lambda_{11} = l(x_0,\ x_1,\ x_5,\ x_8,\ x_{11}) = 3 + 4 + 2 + 5 = 14.$

Etapa a II-a. Formăm diferențele $\lambda_j - \lambda_i$. Se începe cu λ_{11} și se scad din această valoare toate valorile λ_i pentru care $x_i = F^{-1}_{x_{11}}$ (adică λ_i corespunzătoare vîrfurilor adiacente vîrfului x_{11}) obținîndu-se

$$\begin{array}{l}
\lambda_{11} - \lambda_{10} = -7 < l(x_{10},\ x_{11}) = 6 \\
\lambda_{11} - \lambda_9 = -1 < l(x_9,\ x_{11}) = 4 \\
\lambda_{11} - \lambda_8 = 4 < l(x_8,\ x_{11}) = 5
\end{array}$$

$$\begin{array}{l}
\lambda_{10} - \lambda_7 = 8 = l(x_7,\ x_{10}) = 8 \\
\lambda_{10} - \lambda_8 = 11 > l(x_7,\ x_{10}) = 7
\end{array}$$

$$\begin{array}{l}
\lambda_9 - \lambda_6 = 6 = l(x_6,\ x_9) = 6 \\
\lambda_9 - \lambda_8 = 5 > l(x_8,\ x_9) = 1
\end{array}$$

$$\begin{array}{l}
\lambda_8 - \lambda_2 = 9 = l(x_2,\ x_8) = 9 \\
\lambda_8 - \lambda_5 = 3 = l(x_5,\ x_8) = 2 \\
\lambda_8 - \lambda_6 = 1 < l(x_6,\ x_8) = 8
\end{array}$$

$$\begin{array}{l}
\lambda_7 - \lambda_4 = 8 > l(x_4,\ x_7) = 5 \\
\lambda_7 - \lambda_5 = 6 = l(x_5,\ x_7) = 6
\end{array}$$

$$\begin{array}{l}
\lambda_6 - \lambda_2 = 8 > l(x_2,\ x_6) = 6 \\
\lambda_6 - \lambda_3 = 4 = l(x_3,\ x_6) = 4
\end{array}$$

$$\begin{array}{l}
\lambda_5 - \lambda_1 = 4 = l(x_1,\ x_5) = 4 \\
\lambda_5 - \lambda_2 = 6 > l(x_2,\ x_5) = 3
\end{array}$$

$$\lambda_4 - \lambda_1 = 2 = l(x_1,\ x_4) = 2$$

Scanned with OKEN Scanner

$$\lambda_3 - \lambda_0 = 5 = l(x_0, x_3) = 5$$

$$\lambda_2 - \lambda_0 = 1 = l(x_0, x_2) = 1$$

$$\lambda_1 - \lambda_0 = 3 = l(x_0, x_1) = 3$$

Avem de corectat șase diferențe de forma $\lambda_j - \lambda_i > l(x_i, x_j)$, în sensul că înlocuim pe λ_j prin:

$$\begin{aligned}
\lambda'_{10} &= \lambda_8 + l(x_7, x_{10}) = 10 + 8 = 18,\\
\lambda'_9 &= \lambda_8 + l(x_8, x_9) = 10 + 1 = 11,\\
\lambda'_8 &= \lambda_5 + l(x_5, x_8) = 7 + 2 = 9,\\
\lambda'_7 &= \lambda_4 + l(x_4, x_7) = 5 + 5 = 10,\\
\lambda'_6 &= \lambda_2 + l(x_2, x_6) = 1 + 6 = 7,\\
\lambda'_5 &= \lambda_2 + l(x_2, x_5) = 1 + 3 = 4.
\end{aligned}$$

Celelalte valori λ_j rămînînd neschimbate, trecem la recalcularea diferențelor $\lambda'_j - \lambda'_i$ pentru șirul de valori

$$\begin{aligned}
\lambda'_0 &= \lambda_0 = 0,\\
\lambda'_1 &= \lambda_1 = 3,\\
\lambda'_2 &= \lambda_2 = 1,\\
\lambda'_3 &= \lambda_3 = 5,\\
\lambda'_4 &= \lambda_4 = 5,\\
\lambda'_5 &= 4,\\
\lambda'_6 &= 7,\\
\lambda'_7 &= 10,\\
\lambda'_8 &= 9,\\
\lambda'_9 &= 11,\\
\lambda'_{10} &= 18,\\
\lambda'_{11} &= \lambda_{11} = 14,\\
\lambda'_{11} - \lambda'_{10} &= 6 = l(x_{10}, x_{11})\\
\lambda'_{11} - \lambda'_9 &= 3 < l(x_9, x_{11})\\
\lambda'_{11} - \lambda'_8 &= 5 = l(x_8, x_{11})
\end{aligned}$$

$$\begin{aligned}
\lambda'_{10} - \lambda'_7 &= 8 = l(x_7, x_{10})\\
\lambda'_{10} - \lambda'_8 &= 9 > l(x_8, x_{10}) = 7
\end{aligned}$$

$$\begin{aligned}
\lambda'_9 - \lambda'_6 &= 4 < l(x_6, x_9)\\
\lambda'_9 - \lambda'_8 &= 2 > l(x_8, x_9) = 1
\end{aligned}$$

$$\begin{aligned}
\lambda'_8 - \lambda'_2 &= 8 < l(x_2, x_8)\\
\lambda'_8 - \lambda'_5 &= 5 > l(x_5, x_8) = 2\\
\lambda'_8 - \lambda'_6 &= 2 < l(x_6, x_8)
\end{aligned}$$

Scanned with OKEN Scanner

$$\lambda'_7 - \lambda'_4 = 5 = l(x_4, x_7)$$
$$\lambda'_7 - \lambda'_5 = 6 = l(x_5, x_7)$$

$$\lambda'_6 - \lambda'_2 = 6 = l(x_2, x_6)$$
$$\lambda'_6 - \lambda'_3 = 2 < l(x_3, x_6),$$

$$\lambda'_5 - \lambda'_1 = 1 = l(x_1, x_5)$$
$$\lambda'_5 - \lambda'_2 = 3 = l(x_2, x_5)$$

$$\lambda'_4 - \lambda'_1 = 2 = l(x_1, x_4)$$

$$\lambda'_3 - \lambda'_0 = 5 = l(x_0, x_3)$$

$$\lambda'_2 - \lambda'_0 = 1 = l(x_0, x_2)$$

$$\lambda'_1 - \lambda'_0 = 3 = l(x_0, x_1)$$

De această dată avem de corectat trei diferențe de forma $\lambda'_j - \lambda'_i > l(x_i, x_j)$ și anume:

$$\lambda''_{10} = \lambda'_8 + l(x_8, x_{10}) = 9 + 7 = 16,$$
$$\lambda''_9 = \lambda'_8 + l(x_8, x_9) = 9 + 1 = 10,$$
$$\lambda''_8 = \lambda'_5 + l(x_5, x_8) = 4 + 2 = 6.$$

Celelalte valori rămîn neschimbate, adică:

$$\lambda''_0 = 0, \quad \lambda''_4 = 5, \quad \lambda''_8 = 6,$$
$$\lambda''_1 = 3, \quad \lambda''_5 = 4, \quad \lambda''_9 = 10,$$
$$\lambda''_2 = 1, \quad \lambda''_6 = 7, \quad \lambda''_{10} = 16,$$
$$\lambda''_3 = 5, \quad \lambda''_7 = 10, \quad \lambda''_{11} = 14.$$

Calculăm din nou diferențele $\lambda''_j - \lambda''_i$ și avem:

$$\lambda''_{11} - \lambda''_{10} = -2 < l(x_{10}, x_{11}),$$
$$\lambda''_{11} - \lambda''_9 = 4 = l(x_9, x_{11}),$$
$$\lambda''_{11} - \lambda''_8 = 8 > l(x_8, x_{11}) = 5,$$

$$\lambda''_{10} - \lambda''_7 = 6 < l(x_7, x_{10}),$$
$$\lambda''_{10} - \lambda''_8 = 10 > l(x_8, x_{10}) = 7,$$

$$\lambda''_9 - \lambda''_6 = 3 < l(x_6, x_9),$$
$$\lambda''_9 - \lambda''_8 = 4 > l(x_8, x_9) = 1,$$

$$\lambda''_8 - \lambda''_2 = 5 < l(x_2, x_8),$$
$$\lambda''_8 - \lambda''_5 = 2 = l(x_5, x_8),$$
$$\lambda''_8 - \lambda''_6 = -1 < l(x_6, x_8),$$

Scanned with OKEN Scanner

$$\lambda_7'' - \lambda_4'' = 5 = l(x_4, x_7),$$
$$\lambda_7'' - \lambda_5'' = 6 = l(x_5, x_7),$$

$$\lambda_6'' - \lambda_2'' = 6 = l(x_2, x_6),$$
$$\lambda_6'' - \lambda_3'' = 2 < l(x_3, x_6),$$

$$\lambda_5'' - \lambda_1'' = 2 < l(x_1, x_5),$$
$$\lambda_5'' - \lambda_2'' = 3 = l(x_2, x_4),$$

$$\lambda_4'' - \lambda_1'' = 2 = l(x_1, x_4)$$

$$\lambda_3'' - \lambda_0'' = 5 = l(x_0, x_3),$$

$$\lambda_2'' - \lambda_0'' = 1 = l(x_0, x_2)$$

$$\lambda_1 - \lambda_0 = 3 = l(x_0, x_1)$$

Corectăm diferențele $\lambda_j'' - \lambda_i'' > l(x_i, x_j)$;

$$\lambda_{11}^{(3)} = \lambda_8'' + l(x_8, x_{11}) = 6 + 5 = 11,$$
$$\lambda_{10}^{(3)} = \lambda_8'' + l(x_8, x_{10}) = 6 + 7 = 13,$$
$$\lambda_9^{(3)} = \lambda_8'' + l(x_8, x_9) = 6 + 1 = 7.$$

Celelalte valori rămînînd neschimbate, avem la etapa a III-a:

$$\lambda_0^{(3)} = 0, \quad \lambda_4^{(3)} = 5, \quad \lambda_8^{(3)} = 6,$$
$$\lambda_1^{(3)} = 3, \quad \lambda_5^{(3)} = 4, \quad \lambda_9^{(3)} = 7,$$
$$\lambda_2^{(3)} = 1, \quad \lambda_6^{(3)} = 7 \quad \lambda_{10}^{(3)} = 13,$$
$$\lambda_3^{(3)} = 5, \quad \lambda_7^{(3)} = 10, \quad \lambda_{11}^{(3)} = 11.$$

Calculînd diferențele $\lambda_j^{(3)} - \lambda_i^{(3)}$ obținem:

$$\lambda_{11}^{(3)} - \lambda_{10}^{(3)} = -2 < l(x_{10}, x_{11}),$$
$$\lambda_{11}^{(3)} - \lambda_9^{(3)} = 4 = l(x_9, x_{11}),$$
$$\lambda_{11}^{(3)} - \lambda_8^{(3)} = 5 = l(x_8, x_{11}),$$

$$\lambda_{10}^{(3)} - \lambda_7^{(3)} = 3 < l(x_7, x_{10}).$$
$$\lambda_{10}^{(3)} - \lambda_8^{(3)} = 7 = l(x_8, x_{10})$$
$$\lambda_9^{(3)} - \lambda_6^{(3)} = 0 < l(x_6, x_9)$$

$$\lambda_9^{(3)} - \lambda_8^{(3)} = 1 = l(x_8, x_9)$$

$$\lambda_8^{(3)} - \lambda_2^{(3)} = 5 < l(x_2, x_8)$$
$$\lambda_8^{(3)} - \lambda_5^{(3)} = 2 = l(x_5, x_8)$$
$$\lambda_8^{(3)} - \lambda_6^{(3)} = -1 < l(x_6, x_8)$$

Scanned with OKEN Scanner

$$\lambda_7^{(3)} - \lambda_4^{(3)} = 5 = l(x_4, x_7)$$
$$\lambda_7^{(3)} - \lambda_5^{(3)} = 6 = l(x_5, x_7)$$

$$\lambda_6^{(3)} - \lambda_2^{(3)} = 6 = l(x_2, x_6)$$
$$\lambda_6^{(3)} - \lambda_3^{(3)} = 2 \quad l(x_3, x_6)$$

$$\lambda_5^{(3)} - \lambda_1^{(3)} = 1 < l(x_1, x_5)$$
$$\lambda_5^{(3)} - \lambda_1^{(3)} = 3 = l(x_2, x_5)$$

$$\lambda_4^{(3)} - \lambda_1^{(3)} = 2 = l(x_1, x_4)$$

$$\lambda_3^{(3)} - \lambda_0^{(3)} = 5 = l(x_0, x_3)$$

$$\lambda_2^{(3)} - \lambda_0^{(3)} = 1 = l(x_0, x_2)$$

$$\lambda_1^{(3)} - \lambda_0^{(3)} = 3 = l(x_0, x_1)$$

Deoarece toate diferențele obținute în această etapă verifică relația

$$\lambda_j^{(3)} - \lambda_i^{(3)} \leqslant l(x_1, x_j),$$

algoritmul ia sfîrșit.

În vederea alegerii drumului de lungime minimă să fixăm atenția asupra diferențelor de forma: $\lambda_j^{(3)} - \lambda_i^{(3)} = l(x_i, x_j)$, care în exemplul nostru sînt:

$$\lambda_{11}^{(3)} - \lambda_9^{(3)} = 4 \qquad \lambda_6^{(3)} - \lambda_2^{(3)} = 6$$
$$\lambda_{11}^{(3)} - \lambda_8^{(3)} = 5 \qquad \lambda_5^{(3)} - \lambda_2^{(3)} = 3$$
$$\lambda_{10}^{(3)} - \lambda_8^{(3)} = 7 \qquad \lambda_4^{(3)} - \lambda_1^{(3)} = 2$$
$$\lambda_9^{(3)} - \lambda_8^{(3)} = 1 \qquad \lambda_3^{(3)} - \lambda_0^{(3)} = 5$$
$$\lambda_8^{(3)} - \lambda_5^{(3)} = 2 \qquad \lambda_2^{(3)} - \lambda_0^{(3)} = 1$$
$$\lambda_7^{(3)} - \lambda_4^{(3)} = 6 \qquad \lambda_1^{(3)} - \lambda_0^{(3)} = 3$$

Cu ajutorul egalităților putem determina succesiunea de vîrfuri care intră în componența drumului minim.

Orientîndu-ne după indici, avem următoarele drumuri minime:

$$d^{(1)} = (x_0, x_2, x_5, x_8, x_9, x_{11})$$

și

$$d^{(2)} = (x_0, x_2, x_5, x_8, x_{11}).$$

Scanned with OKEN Scanner

Succesiunile de vîrfuri care constituie drumurile minime au fost găsite astfel: $\lambda_{11} - \lambda_9 = 4$, care indică existența în drum a vîrfurilor x_{11} și x_9; apoi am trecut la o diferență care are ca descăzut pe λ_9, adică $\lambda_9 - \lambda_8 = 1$, și aceasta pentru a face legătura între vîrful x_9 și vîrful următor al drumului. Tot așa am găsit și vîrfurile x_5, x_2, x_0.

Deoarece pentru vîrful x_{11} există două diferențe pe care le putem lua în considerație, rezultă că există două soluții optime ale problemei (două drumuri cu aceeași lungime minimă).

Urmărind succesiunea de arce care intră în componența fiecărui drum optim, se poate constata cu ușurință că lungimea minimă este dată de $\lambda_{11} = 11$. Într-adevăr,

$$l(x_0, x_2, x_5, x_8, x_9, x_{11}) = 1 + 3 + 2 + 1 + 4 = 11,$$
$$l(x_0, x_2, x_5, x_8, x_{11}) = 1 + 3 + 2 + 5 = 11.$$

Din punct de vedere economic, existența soluției multiple prezintă un mare interes, oferind posibilitatea alegerii, după nevoie, a unuia sau a celuilalt drum.

Astfel, dacă vom pune problema construirii unei șosele care să treacă prin localitățile x_0, x_2, x_5, x_8 și există posibilitatea de a merge pe două trasee diferite, firește că se va alege acel traseu pentru care condițiile de lucru (teren, apropiere de centrele de aprovizionare etc) sînt mai avantajoase.

Analizînd modul în care prin algoritmul lui Ford se caută un drum de lungime minimă, se pot deduce următoarele: pentru ca lungimea totală a drumului să fie minimă, trebuie ca acesta să conțină, pe cît posibil, arce de lungimi cît mai mici.

Pornind de la această observație și pe baza teoremei lui Bellman, autorii au elaborat următorul algoritm:

2°. Fie dat graful $g = (X,F)$ finit și fără circuite.

a) Se caută arcul de lungime minimă $a \in A$, a cărui lungime se scade din lungimile tuturor arcelor. În acest mod se vor obține arce de lungime zero, acolo unde acestea au lungimea egală cu aceea a arcului minim.

b) Astfel de arce vor fi eliminate din graful respectiv, obținînd graful parțial $G_1 = (X, F_1)$. În G_1 alegem din nou arcul de lungime minimă $a_1 \in A_1$, pe care-l scădem din toate

arcele grafului G_1, obținînd în acest fel iarăși arce de lungime zero corespunzătoare arcului a_1 și celorlalte arce cu aceeași lungime.

În graful G_1 vom elimina din nou toate arcele de lungime zero, obținînd graful parțial $G_2 = (X, F_2)$.

Repetăm procedeul arătat, pînă cînd vom ajunge la un moment dat să obținem arce de lungime zero aparținînd unor drumuri care duc de la x_0 la x_n.

c) Să trecem la determinarea drumului minim prin selectarea arcelor de lungime zero. De această dată, în componența drumului avem grijă să alegem, pe cît posibil, arce de lungime zero făcînd parte din prima iterație, apoi din a doua etc., pînă cînd se ajunge în vîrful x_n.

Deoarece existența arcelor de lungime zero alese presupune arce de lungimi cît mai mici, sîntem siguri că drumul căutat este cel minim.

Pentru ilustrare, să reluăm exemplul din figura II.3.3 pe care îl tratăm în etape.

Deoarece s-au găsit drumuri cu arce de lungime zero de la x_0 la x_{11}, algoritmul se termină.

Arcele care intră în componența drumului de lungime minimă sînt: (x_0, x_2), (x_2, x_5), (x_5, x_8), (x_8, x_9), (x_9, x_{11}).

Evident că vom lua în considerație acel drum în componența căruia intră cît mai multe arce de valoare zero, din primele etape: în cazul nostru, drumul căutat este: (x_0, x_2), (x_2, x_5), (x_5, x_8), (x_8, x_9), (x_9, x_{11}).

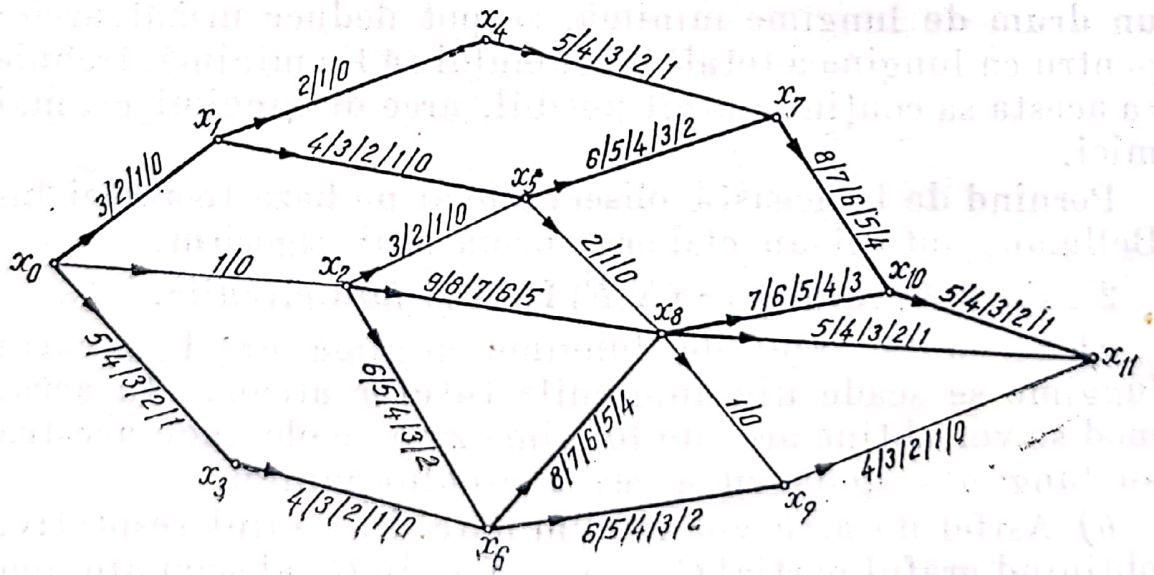


Fig. II.3.3

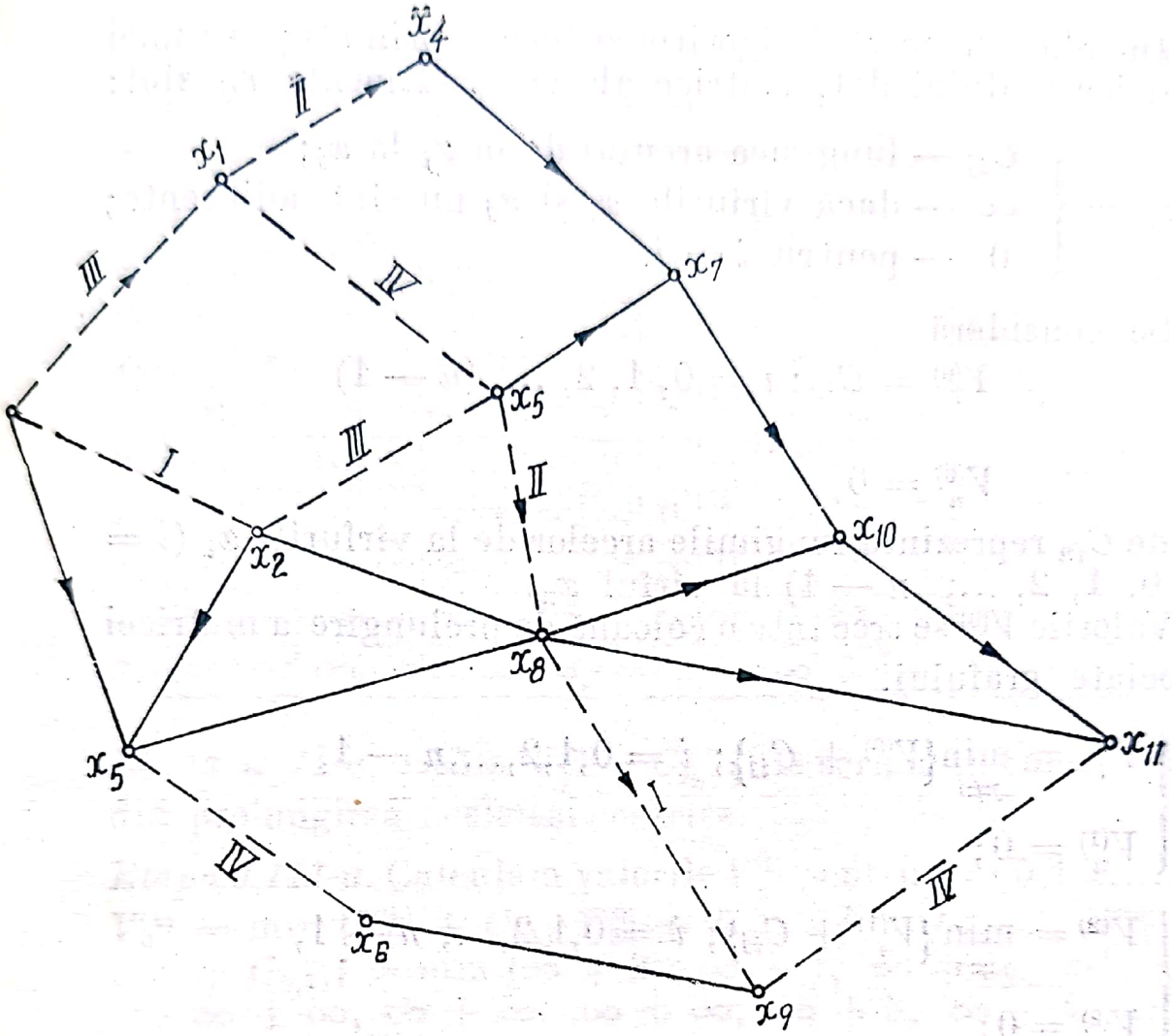


Fig. II.3.4

Graful din figura II.3.4 arată succesiunea de arce a drumului minim.

Pentru a calcula lungimea minimă a drumului de la x_0 la x_{11}, vom însuma ponderile inițiale, corespunzătoare acestor arce.

Avem

$$l(d_{\min}) = l(x_0, x_2) + l(x_2, x_5) + l(x_5, x_8) + l(x_8, x_9) + \\ + l(x_9, x_{11}) = 1 + 3 + 2 + 1 + 4 = 11$$

3°. Algoritmul Bellman-Kalaba. Pentru determinarea drumului de lungime minimă într-un graf, se folosește și algoritmul Bellman-Kalaba, care se deosebește de algoritmul lui Ford, prin faptul că ordonează calculele și permite programarea lui pe mașinile electronice de calcul.

Scanned with OKEN Scanner

În aplicarea acestui algoritm se începe prin ataşarea unei matrice grafului dat, matrice ale cărei elemente r_{ij} sînt:

$$r_{ij} = \begin{cases} C_{ij} \text{ — lungimea arcului de la } x_i \text{ la } x_j; \\ \infty \text{ — dacă vîrfurile } x_i \text{ şi } x_j \text{ nu sînt adiacente;} \\ 0 \text{ — pentru } i = j. \end{cases}$$

Se consideră

$$V_i^{(0)} = C_{in};\ i = 0, 1, 2, \ldots, (n-1)$$

şi

$$V_n^{(0)} = 0,$$

unde C_{in} reprezintă lungimile arcelor de la vîrfurile x_i ($i = 0, 1, 2, \ldots, n-1$) la vîrful x_n.

Valorile $V_j^{(0)}$ se trec într-o coloană de prelungire a matricei asociate grafului.

$$\begin{cases} V_i^{(1)} = \min\limits_{j \neq i} \{V_i^{(0)} + C_{ij}\};\ i = 0,1,2,\ldots,n-1, \\ V_n^{(1)} = 0; \end{cases}$$

$$\begin{cases} V_i^{(2)} = \min\limits_{j \neq i} \{V_j^{(1)} + C_{ij}\};\ i = 0,1,2,\ldots,n-1, \\ V_n^{(2)} = 0; \end{cases}$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$\begin{cases} V_i^{(k)} = \min\limits_{j \neq i} \{V_j^{(1-1)} + C_{ij}\};\ i = 0,1,2,\ldots,n-1, \\ V_n^{(k)} = 0; \end{cases}$$

$$\begin{cases} V_i^{(k+1)} = \min\limits_{j \neq i} \{V_j^{(k)} + C_{ij}\};\ i = 0,1,2,\ldots,n-1, \\ V_n^{(k+1)} = 0. \end{cases}$$

Dacă coloanele $(V_j^{(k)})$ şi $(V_j^{(k+1)})$ coincid, adică dacă $V_j^{(k+1)} = V_j^{(k)}$ ($i = 0,1,2,\ldots, n$), algoritmul ia sfîrşit, deoarece în drumul minim au fost alese drumurile parţiale de lungime minimă.

Să aplicăm acest algoritm pe exemplul din figura II.2.2. *Etapa I*. Matricea asociată grafului din tabelul următor:

Scanned with OKEN Scanner

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	$V_j^{(0)}$	$V_j^{(1)}$	$V_j^{(2)}$	$V_j^{(3)}$	$V_j^{(4)}$
x_0	0	3	1*	5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	15	11	11
x_1	∞	0	∞	∞	2	4*	∞	∞	∞	∞	∞	∞	∞	∞	11	11	11
x_2	∞	∞	0	∞	∞	3*	6	∞	9	∞	∞	∞	∞	14	10	10	10
x_3	∞	∞	∞	0	∞	∞	4*	∞	∞	∞	∞	∞	∞	∞	14	14	14
x_4	∞	∞	∞	∞	0	∞	∞	5*	∞	∞	∞	∞	∞	∞	19	19	19
x_5	∞	∞	∞	∞	∞	0	∞	6	2*	∞	∞	∞	∞	7	7	7	7
x_6	∞	∞	∞	∞	∞	∞	0	∞	8	6*	∞	∞	∞	10	10	10	10
x_7	∞	∞	∞	∞	∞	∞	∞	0	∞	∞	8*	∞	∞	14	14	14	14
x_8	∞	∞	∞	∞	∞	∞	∞	∞	0	1	7	5*	5	5	5	5	5
x_9	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	∞	4*	4	4	4	4	4
x_{10}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	6*	6	6	6	6	6
x_{11}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	0	0	0	0	0

Etapa a II-a. Luăm $V_j^{(0)} = C_{i_n}$ și le scriem în coloana $V_j^{(0)}$ din prelungirea aceleiași matrice.

Etapa a III-a. Calculăm valorile $V_i^{(1)}$ pentru $i = 0,1,2,\ldots,10$

$V_0^{(1)} = \min\{V_1^{(0)} + C_{01},\ V_2^{(0)} + C_{02},\ V_3^{(0)} + C_{03},\ldots,\ V_{11}^{(0)} + C_{0,11}\} = \min\{\infty + 3,\ \infty + 1,\ \infty + 5,\ \infty + \infty,\ \infty + \infty,\ \infty + \infty,\ \infty + \infty,\ \infty + 5,\ \infty + 4,\ \infty + 6,\ \infty + 0\} = \infty;$

$V_1^{(1)} = \min\{V_0^{(0)} + C_{10},\ V_2^{(0)} + C_{12},\ V_3^{(0)} + C_{13},\ \ldots,\ V_{11}^{(0)} + C_{1,11}\} = \min\{\infty + \infty,\ \infty + \infty,\ \infty + \infty,\ \infty + 2,\ \infty + 4,\ \infty + \infty,\ \infty + 5,\ \infty + 4,\ \infty + 6,\ \infty + 0\} = \infty;$

$V_2^{(1)} = \min\{V_0^{(0)} + C_{20},\ V_1^{(0)} + C_{21},\ V_3^{(0)} + C_{23},\ldots,\ V_{11}^{(0)} + C_{2,11}\} = \min\{\infty + \infty,\ \infty + \infty,\ \infty + \infty,\ \infty + \infty,\ \infty + 3,\ \infty + 6,\ \infty + \infty,\ 9 + 5,\ \infty + 4,\ \infty + 6,\ \infty + 0\} = 14;$

$V_3^{(1)} = \min\{V_0^{(0)} + C_{30},\ V_1^{(0)} + C_{31},\ V_2^{(0)} + C_{32},\ \ldots,\ V_{11}^{(0)} + C_{3,11}\} = \min\{\infty + \infty,\ \infty + \infty,\ \infty + \infty,\ \infty + \infty,\ \infty + \infty,\ \infty + 4,\ \infty + \infty,\ \infty + 5,\ \infty + 4,\ \infty + 6,\ \infty + 0\} = \infty;$

$V_4^{(1)} = \min\{V_0^{(0)} + C_{40},\ V_1^{(0)} + C_{41},\ V_2^{(0)} + C_{42},\ \ldots,\ V_{11}^{(0)} + C_{4,11}\} = \min\{\infty + \infty,\ \infty + \infty,\ \infty + \infty,\ \infty + \infty,\ \infty + \infty,\ \infty + \infty,\ \infty + 5,\ \infty + 5,\ \infty + 4,\ \infty + 6,\ \infty + 0\} = \infty;$

Scanned with OKEN Scanner

$V_5^{(1)} = \min \{V_0^{(0)} + C_{50},\ V_1^{(0)} + C_{51},\ V_2^{(0)} + C_{52},\ \ldots,\ V_{11}^{(0)} + C_{5,11}\} = \min \{\infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + 6, 5 + 2, 4 + \infty, 6 + \infty, 0 + \infty\} = 7;$

$V_6^{(1)} = \min \{V_0^{(0)} + C_{60},\ V_1^{(0)} + C_{61},\ V_2^{(0)} + C_{62},\ \ldots,\ V_{11}^{(0)} + C_{6,11}\} = \min \{\infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, 8 + 5, 6 + 4, \infty + 6, \infty + 0\} = 10;$

$V_7^{(1)} = \min \{V_0^{(0)} + C_{70},\ V_1^{(0)} + C_{71},\ V_2^{(0)} + C_{72},\ \ldots,\ V_{11}^{(0)} + C_{7,11}\} = \min \{\infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + 5, \infty + 4, 8 + 6, \infty + 0\} = 14;$

$V_8^{(0)} = \min \{V_0^{(0)} + C_{80},\ V_1^{(0)} + C_{81},\ V_2^{(0)} + C_{82}, \ldots,\ V_{11}^{(0)} + C_{8,11}\} = \min \{\infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + 7, \infty + \infty, \infty + \infty, 4 + 1, 7 + 6, 5 + 0\} = 5;$

$V_9^{(1)} = \min \{V_0^{(0)} + C_{90},\ V_1^{(0)} + C_{91},\ V_2^{(0)} + C_{92},\ \ldots,\ V_{11}^{(0)} + C_{9,11}\} = \min \{\infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + 5, \infty + 6, 4 + 0\} = 4;$

$V_{10}^{(1)} = \min \{V_0^{(0)} + C_{10,0},\ V_1^{(0)} + C_{10,1},\ \ldots,\ V_{11}^{(0)} + C_{10,11}\} = \min \{\infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, \infty + \infty, 5 + \infty, 4 + 8, 6 + 0\} = 6$

și

$V_{11}^{(1)} = 0.$

Aceste rezultate le trecem pe coloana $V_j^{(1)}$ din tabel.

Calculăm valorile $V_j^{(2)}$, astfel:

$V_0^{(2)} = \min \{V_1^{(1)} + C_{01},\ V_2^{(1)} + C_{02},\ \ldots,\ V_{11}^{(1)} + C_{0,11}\} = \min \{3 + \infty, 1 + 14, 5 + \infty, \infty + \infty, \infty + 7, \infty + 10, \infty + 14, \infty + 5, \infty + 4, \infty + 6, \infty + 0\} = 15$

$V_1^{(2)} = \min \{V_0^{(1)} + C_{10},\ V_1^{(1)} + C_{11},\ V_2^{(1)} + C_{12},\ \ldots,\ V_{11}^{(1)} + C_{1,11}\} = \min \{\infty + \infty, \infty + 14, \infty + \infty, \infty + 2, 7 + 4, \infty + 10, \infty + 14, \infty + 5, \infty + 4, \infty + 6, \infty + 0\} = 11.$

La fel au fost calculate și celelalte valori de pe coloana $V_i^{(2)}$.

Pentru calculul valorilor corespunzătoare coloanei $V^{(3)}$ procedăm analog:

$V_0^{(3)} = \min \{V_1^{(2)} + C_{01},\ V_2^{(2)} + C_{02},\ \ldots,\ V_{11}^{(2)} + C_{011}\} =$

Scanned with OKEN Scanner

$$= \min \{3 + 11, 10 + 1, 5 + 14, \infty + 19, \infty + 7, \infty + \\ + 10, \infty + 14, \infty + 5, \infty + 4, \infty + 6, \infty + 0\} = 11;$$

$$V_1^{(3)} = \min \{V_0^{(2)} + C_{10}, V_2^{(2)} + C_{12}, \ldots, V_{11}^{(2)} + C_{1,11}\} = \\ = \min \{\infty + 15, \infty + 10, \infty + 14, 2 + 19, 4 + 7, \infty + \\ + 10, \infty + 14, \infty + 15, \infty + 4, \infty + 6, \infty + 0\} = 11;$$

și așa mai departe, se calculează valorile $V_i^{(4)}$ și se constată că ele sînt egale cu valorile din coloana $V_i^{(3)}$.

Lungimea drumului este dată de prima valoare din coloana $V_i^{(4)}$ sau $V_i^{(3)}$, adică

$$\min l(x_0, x_{11}) = V_0^{(3)} = V_0^{(4)} = 11.$$

Observații:

a) Pe coloanele $V_j^{(3)}$ și $V_j^{(4)}$ constatăm că avem două drumuri distincte cu aceeași lungime (11), lucru care s-a scos în evidență și cînd s-a aplicat algoritmul lui Ford.

b) Celelalte valori de pe aceleași coloane $V_j^{(3)}$ și $V_j^{(4)}$ ne arată lungimile drumurilor minime de la fiecare vîrf al grafului la vîrful x_{11}. Astfel, valoarea (10) imediat următoare valorii minime (11) reprezintă lungimea cea mai mică a drumului ce merge de la x_2 la x_{11}.

c) Pentru stabilirea succesiunii de vîrfuri care formează drumul căutat, urmărim pe coloana $V_j^{(4)}$ sau $V_j^{(3)}$ valorile strict descrescătoare ce ne conduc de la x_{11} la fiecare vîrf x_i. Astfel, avem posibilitatea să alegem în componența drumurilor minime vîrfurile $x_0, x_2, x_5, x_8, x_9, x_{11}$:

$$l(d_1) = l(x_0, x_2, x_5, x_8, x_{11}) = 11,$$
$$l(d_2) = l(x_0, x_2, x_5, x_8, x_9, x_{11}) = 11.$$

Oricare alt drum dintre vîrfurile x_0 și x_{11} are lungimea mai mare decît 11.

§ 4. DRUM DE LUNGIME MAXIMĂ ÎNTR-UN GRAF

După cum am mai arătat, practica impune să se găsească uneori și lungimea maximă a drumului dintre două vîrfuri ale grafului.

Scanned with OKEN Scanner

Așa, de exemplu, dacă graful dat reprezintă un proiect de executare a unei lucrări complexe, numerele C_{ij} atașate fiecărui arc reprezintă timpul necesar pentru a efectua o lucrare ce se desfășoară între momentul x_i și x_j. În asemenea situații, sîntem interesați să precizăm care va fi timpul maxim necesar pentru executarea lucrării.

Vom arăta cum algoritmul lui Ford, modificat corespunzător, ne permite să găsim drumul de lungime maximă.

1°. Algoritmul lui Ford pentru găsirea drumului de lungime maximă. Am văzut că pentru determinarea drumului de lungime minimă se impune atașarea unor valori λ_i, ($i = 0,1,2,\ldots n$) fiecărui vîrf x_i al grafului, valori care reprezintă lungimile drumurilor de la x_0 la fiecare vîrf x_i. Această operație rămîne valabilă și în cazul în care se cercetează drumul de lungime maximă; numai că, de această dată, este recomandabil ca valorile λ_i să corespundă unor drumuri parțiale de lungime maximă.

În etapa a doua calculăm de asemenea aceleași diferențe $\lambda_j - \lambda_i$, pe care le comparăm cu lungimile arcelor, $l(x_i, x_j)$.

Din aceste comparații vor rezulta situațiile cunoscute:

$$\lambda_j - \lambda_i < l(x_i, x_j),$$
$$\lambda_j - \lambda_i = l(x_i, x_j),$$
$$\lambda_j - \lambda_i > l(x_i, x_j).$$

Analiza acestor inegalități se face acum luînd în considerație diferențele de forma

$$\lambda_j - \lambda_i < (x_i, x_j),$$

care se corectează în sensul că valorile λ_j se înlocuiesc cu

$$\lambda'_j = \lambda_i + l(x_i, x_j) > \lambda_j.$$

Operațiile continuă pînă cînd toate diferențele $\lambda_j - \lambda_i$ vor fi egale sau mai mari decît $l(x_j, x_i)$.

Pentru găsirea drumului de lungime maximă ne fixăm atenția asupra egalităților de forma:

$$\lambda_j - \lambda_i = l(x_i, x_j).$$

Scanned with OKEN Scanner

Dacă la o anumită etapă k avem obținute egalitățile:

$$\lambda_n^{(k)} - \lambda_{i_1}^{(k)} = l(x_{i_1}, x_n),$$

$$\lambda_{i_1}^{(k)} - \lambda_{i_2}^{(k)} = l(x_{i_2}, x_{i_1}),$$

$$\ldots\ldots\ldots\ldots\ldots\ldots$$

$$\lambda_{i_p}^{(k)} - \lambda_0^{(k)} = l(x_0, x_{i_p}),$$

reținem și de această dată pe acelea ale căror indici formează un șir descrescător.

Vîrfurile ce vor corespunde numerelor $\lambda_j^{(k)}$ din acest șir vor intra în structura drumului căutat.

Observații:

a) Menționăm că și în cazul drumului de lungime maximă pot exista drumuri multiple, după cum valoarea $\lambda_j^{(k)}$ apare de mai multe ori în șirul monoton descrescător,

b) Nu trebuie pierdut din vedere că graful pentru care se determină drumul de lungime maximă trebuie să fie fără circuite, deoarece în caz contrar el conține drumuri nemărginite.

Pentru înțelegerea modului în care se aplică algoritmul lui Ford, în cazul în care este vorba de determinarea drumului maxim, să cercetăm drumul de lungime maximă dintre vîrful x_0 și vîrful x_{13} al grafului din figura II.4.1.

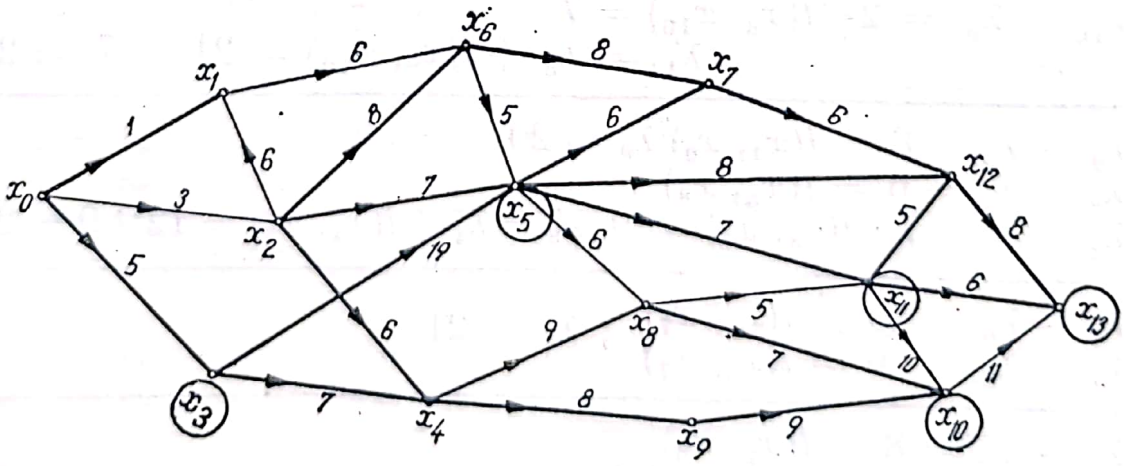


Fig. II.4.1

Etapa I. Asociem fiecărui vîrf x_i un număr $\lambda_i (i = 0,1,\ldots, 13)$, care reprezintă drumul de la x_0 la x_i, iar vîrfului x_0 îi asociem

$\lambda_0 = 0.$ $\lambda_5 = 15$ (prin x_3) $\lambda_{10} = 23$ (prin x_2, x_5, x_8)
$\lambda_1 = 9$ (prin x_2) $\lambda_6 = 11$ (prin x_2) $\lambda_{11} = 17$ (prin x_2 și x_5)
$\lambda_2 = 3$ $\lambda_7 = 21$ (prin x_3, x_5) $\lambda_{12} = 27$ (prin x_3, x_5, x_{11})
$\lambda_3 = 5$ $\lambda_8 = 21$ (prin x_3, x_4) $\lambda_{13} = 40$ (prin x_3, x_4, x_9, x_{10})
$\lambda_4 = 12$ (prin x_3) $\lambda_9 = 20$ (prin x_3, x_4)

Formăm diferențele $\lambda_j - \lambda_i$

$\lambda_{13} - \lambda_{12} = 13 > l(x_{12}, x_{13})$
$\lambda_{13} - \lambda_{11} = 23 > l(x_{11}, x_{13})$
$\lambda_{13} - \lambda_{10} = 17 > l(x_{10}, x_{13})$ $\quad \lambda'_{13} = 40$

$\lambda_{12} - \lambda_{11} = 10 > l(x_{11}, x_{12})$
$\lambda_{12} - \lambda_5 = 12 > l(x_5, x_{12})$ $\quad \lambda'_{12} = 27$
$\lambda_{12} - \lambda_7 = 6 = l(x_7, x_{12})$

$\lambda_{11} - \lambda_8 = -4 < l(x_8, x_{11}) = 5;$
$\lambda'_{11} = \lambda_8 + l(x_8, x_{11}) = 21 + 5 = 26$
$\lambda_{11} - \lambda_5 = 2 < l(x_5, x_{11}) = 7;$
$\lambda'_{11} = \lambda_5 + l(x_5, x_{11}) = 15 + 7 = 22$

$\lambda_{10} - \lambda_{11} = 6 < l(x_{10}, x_{11}) = 10;$
$\lambda'_{10} = \lambda_{11} + l(x_{10}, x_{11}) = 17 + 10 = 27$
$\lambda_{10} - \lambda_9 = 3 < l(x_9, x_{10}) = 9;$
$\lambda'_{10} = \lambda_9 + l(x_9, x_{10}) = 20 + 9 = 29$
$\lambda_{10} - \lambda_8 = 2 < l(x_8, x_{10}) = 7$
$\lambda'_{10} = \lambda_8 + l(x_8, x_{10}) = 21 + 7 = 28$

$\lambda_9 - \lambda_4 = 8 = l(x_4, x_9)$ $\lambda'_9 = 20$
$\lambda_8 - \lambda_5 = 6 = l(x_5, x_8)$
$\lambda_8 - \lambda_4 = 7 < l(x_4, x_8) = 9$ $\lambda'_8 = \lambda_4 + l(x_4, x_8) = 12 + 9 = 21$

$\lambda_7 - \lambda_6 = 10 > l(x_6, x_7)$; $\lambda'_7 = 21$
$\lambda_7 - \lambda_5 = 6 = l(x_5, x_7)$

$\lambda_6 - \lambda_2 = 8 = l(x_2, x_6)$
$\lambda_6 - \lambda_1 = 2 < l(x_1, x_6) = 6;$ $\lambda'_6 = \lambda_1 + l(x_1, x_6) = 9 + 6 = 15$

Scanned with OKEN Scanner

$\lambda_5 - \lambda_6 = 4 < l(x_6, x_5) = 5$; $\lambda'_5 = \lambda_6 + l(x_6, x_5) = 11 + 5 = 16$
$\lambda_5 - \lambda_3 = 10 = l(x_3, x_5)$
$\lambda_5 - \lambda_2 = 12 > l(x_2, x_5)$

$\lambda_4 - \lambda_2 = 9 > l(x_2, x_4)$
$\lambda_4 - \lambda_3 = 7 = l(x_3, x_4)$ $\quad \lambda'_4 = 12$

$\lambda_3 - \lambda_0 = 5 = l(x_0, x_5)$ $\quad \lambda'_3 = 5$

$\lambda_2 - \lambda_0 = 3 = l(x_0, x_2)$ $\quad \lambda'_2 = 3$

$\lambda_1 - \lambda_2 = 6 = l(x_1, x_2)$
$\lambda_1 - \lambda_0 = 9 > l(x_0, x_1)$ $\quad \lambda'_1 = 9$

Pentru λ'_{11} și λ'_{10} am găsit mai multe valori corectate. Dintre acestea, vom reține pe acelea care au valoarea cea mai mare și anume:

$$\lambda'_{11} = 26 \text{ și } \lambda'_{10} = 29.$$

Construim din nou diferențele $\lambda'_j - \lambda'_i$, pe care le comparăm cu lungimile $l(x_i, x_j)$:

$\lambda'_{13} - \lambda'_{12} = 13 > l(x_{12}, x_{13})$
$\lambda'_{13} - \lambda'_{11} = 14 > l(x_{11}, x_{13})$
$\lambda'_{13} - \lambda'_{10} = 11 = l(x_{10}, x_{13})$; $\lambda''_{13} = 40$

$\lambda'_{12} - \lambda'_{11} = 1 < l(x_{11}, x_{12}) = 5$; $\lambda''_{12} = \lambda'_{11} + l(x_{11}, x_{12}) = 27 + 5 = 32$
$\lambda'_{12} - \lambda'_5 = 11 > l(x_5, x_{12})$
$\lambda'_{12} - \lambda'_7 = 6 = l(x_7, x_{12})$

$\lambda'_{11} - \lambda'_8 = 5 = l(x_8, x_{11})$; $\lambda''_{11} = 26$
$\lambda'_{11} - \lambda'_5 = 10 > l(x_5, x_{11})$

$\lambda'_{10} - \lambda'_{11} = 3 < l(x_{11}, x_{10}) = 10$; $\lambda''_{10} = \lambda''_{11} + l(x_{11}, x_{10}) = 26 + 10 = 36$
$\lambda'_{10} - \lambda'_9 = 9 = l(x_9, x_{10})$
$\lambda'_{10} - \lambda'_8 = 7 = l(x_8, x_{10})$

$\lambda'_9 - \lambda'_4 = 8 = l(x_4, x_9)$; $\lambda''_9 = 20$

Scanned with OKEN Scanner

$\lambda'_8 - \lambda'_5 = 5 < l(x_5, x_8) = 6$; $\lambda''_8 = \lambda'_5 + l(x_5, x_8) = 16 + 6 = 22$

$\lambda'_8 - \lambda'_4 = 9 = l(x_4, x_8)$

$\lambda'_7 - \lambda'_6 = 6 < l(x_6, x_7) = 7$; $\lambda''_7 = \lambda'_6 + l(x_6, x_7) = 17 + 7 = 22$

$\lambda'_7 - \lambda'_5 = 5 < l(x_5, x_7) = 6$; $\lambda''_7 = \lambda'_5 + l(x_5, x_7) = 16 + 6 = 22$

$\lambda'_6 - \lambda'_2 = 12 > l(x_2, x_6)$; $\lambda''_6 = 15$

$\lambda'_6 - \lambda'_1 = 6 = l(x_1, x_6)$

$\lambda'_5 - \lambda'_6 = 1 < l(x_6, x_5) = 5$

$\lambda'_5 - \lambda'_3 = 11 > l(x_3, x_5)$ $\lambda''_5 = \lambda'_6 + l(x_6, x_5) = 15 + 5 = 20$

$\lambda'_5 - \lambda'_2 = 13 > l(x_2, x_5)$

$\lambda'_4 - \lambda'_2 = 9 > l(x_2, x_4)$ $\lambda''_4 = 12$

$\lambda'_4 - \lambda'_3 = 7 = l(x_3, x_4)$

$\lambda'_3 - \lambda'_0 = 5 = l(x_0, x_3)$ $\lambda''_3 = 5$

$\lambda'_2 - \lambda'_0 = 3 = l(x_0, x_2)$ $\lambda''_2 = 3$

$\lambda'_1 - \lambda'_0 = 9 > l(x_0, x_1)$

$\lambda'_1 - \lambda'_2 = 6 = l(x_1, x_2)$ $\lambda''_1 = 9$

$\lambda''_0 = 0$

Se fac din nou diferențele $\lambda''_j - \lambda''_i$, pe care le comparăm cu lungimile $l(x_i, x_j)$ și obținem:

$\lambda''_{13} - \lambda''_{12} = 8 = l(x_{12}, x_{13})$,

$\lambda''_{13} - \lambda''_{11} = 14 > l(x_{11}, x_{13})$,

$\lambda''_{13} - \lambda''_{10} = 4 < l(x_{10}, x_{13}) = 11$; $\lambda^{(3)}_{13} = \lambda''_{10} + l(x_{10}, x_{13}) = 36 + 11 = 47$

$\lambda''_{12} - \lambda''_{11} = 6 > l(x_{11}, x_{12})$

$\lambda''_{12} - \lambda''_5 = 17 > l(x_5, x_{12})$ $\lambda^{(3)}_{12} = 32$

$\lambda''_{12} - \lambda''_7 = 10 > l(x_7, x_{12})$

$\lambda''_{11} - \lambda'_8 = 4 < l(x_8, x_{11}) = 5$; $\lambda^{(3)}_{11} = \lambda''_8 + l(x_8, x_{11}) = 22 + 5 = 27$

$\lambda''_{11} - \lambda'_5 = 6 < l(x_5, x_{11}) = 7$; $\lambda^{(3)}_{11} = \lambda''_5 + l(x_5, x_{11}) = 20 + 7 = 27$

Scanned with OKEN Scanner

$\lambda''_{10} - \lambda''_{11} = 10 = l(x_{10}, x_{10})$
$\lambda''_{10} - \lambda''_{9} = 16 > l(x_9, x_{10})$; $\lambda^{(3)}_{10} = 36$
$\lambda''_{10} - \lambda''_{8} = 14 > l(x_8, x_{10})$

$\lambda''_9 - \lambda''_4 = 8 = l(x_4; x_9)$; $\lambda^{(3)}_9 = 20$

$\lambda''_8 - \lambda''_5 = 2 < l(x_5, x_8) = 6$; $\lambda^{(3)}_8 = \lambda''_5 + l(x_5, x_8 = 20 + 6 = 26$
$\lambda''_8 - \lambda''_4 = 10 > l(x_4, x_8)$

$\lambda''_7 - \lambda''_6 = 7 = l(x_6, x_7)$
$\lambda''_7 - \lambda''_5 = 2 < l(x_5, x_7) = 6$; $\lambda^{(3)}_7 = \lambda''_5 + l(x_5, x_7) = 20 + 6 = 26$

$\lambda''_6 - \lambda''_2 = 12 > l(x_2, x_6)$; $\lambda^{(3)}_6 = 15$
$\lambda''_6 - \lambda''_1 = 6 = l(x_1, x_6)$

$\lambda''_5 - \lambda''_6 = 5 = l(x_6, x_5)$
$\lambda''_5 - \lambda''_3 = 10 = l(x_3, x_5)$; $\lambda^{(3)}_5 = 20$
$\lambda''_5 - \lambda''_2 = 17 > l(x_2, x_3)$

$\lambda''_4 - \lambda''_2 = 9 > l(x_2, x_4)$
$\lambda''_4 - \lambda''_3 = 7 = l(x_3, x_7)$; $\lambda^{(3)}_4 = 12$

$\lambda''_3 - \lambda''_0 = 5 = l(x_0, x_3)$; $\lambda^{(3)}_3 = 5$

$\lambda''_2 - \lambda''_0 = 3 = l(x_0, x_2)$; $\lambda^{(3)}_2 = 3$

$\lambda''_1 - \lambda''_0 = 9 > l(x_0, x_1)$; $\lambda^{(3)}_1 = 9$

$\lambda''_1 - \lambda''_2 = 6 = l(x_2, x_1)$; $\lambda^{(3)}_1 = 6$ și $\lambda^{(3)}_0 = 0$

Deoarece mai avem încă

$$\lambda^{(3)}_j - \lambda^{(3)}_i < l(x_i, x_j),$$

procedeul continuă:

$\lambda^{(3)}_{13} - \lambda^{(3)}_{12} = 15 > l(x_{12}, x_{13})$; $\lambda^{(4)}_{13} = 47$
$\lambda^{(3)}_{13} - \lambda^{(3)}_{11} = 20 > l(x_{11}, x_{13})$
$\lambda^{(3)}_{10} - \lambda^{(3)}_{10} = 11 = l(x_{10}, x_{13})$

$\lambda^{(3)}_{12} - \lambda^{(3)}_{11} = 5 = l(x_{11}, x_{12})$
$\lambda^{(3)}_{12} - \lambda^{(3)}_{5} = 12 > l(x_5, x_{12})$; $\lambda^{(4)}_{12} = 32$
$\lambda^{(3)}_{12} - \lambda^{(3)}_{7} = 6 = l(x_7, x_{12})$

Scanned with OKEN Scanner

$\lambda_{11}^{(3)} - \lambda_8^{(3)} = 1 < l(x_8, x_{11}) = 5$; $\lambda_{11}^{(4)} = \lambda_8^{(3)} + l(x_1, x_{11}) = 26 + 5 = 31$

$\lambda_{11}^{(3)} - \lambda_5^{(3)} = 7 = l(x_5, x_{11})$

$\lambda_{10}^{(3)} - \lambda_{11}^{(3)} = 9 < l(x_{11}, x_{10}) = 10$; $\lambda_{10}^{(4)} = \lambda_{11}^{(3)} + l(x_{11}, x_{10}) = 27 + 10 = 37$

$\lambda_{10}^{(3)} - \lambda_9^{(3)} = 16 > l(x_9, x_{10})$

$\lambda_{10}^{(3)} - \lambda_8^{(3)} = 10 > l(x_8, x_{10})$

$\lambda_8^{(3)} - \lambda_4^{(3)} = 8 = l(x_4, x_9)$; $\lambda_9^{(4)} = 20$

$\lambda_8^{(3)} - \lambda_5^{(3)} = 6 = l(x_5, x_8)$ $\lambda_8^{(4)} = 26$

$\lambda_8^{(3)} - \lambda_4^{(3)} = 14 > l(x_4, x_8)$

$\lambda_7^{(3)} - \lambda_6^{(3)} = 11 > l(x_6, x_7)$; $\lambda_7^{(4)} = 26$

$\lambda_7^{(3)} - \lambda_5^{(3)} = 6 = l(x_5, x_7)$

$\lambda_6^{(3)} - \lambda_2^{(3)} = 12 > l(x_2, x_6)$; $\lambda_6^{(4)} = 15$

$\lambda_6^{(3)} - \lambda_1^{(3)} = 6 = l(x_1, x_6)$

$\lambda_5^{(3)} - \lambda_6^{(3)} = 5 = l(x_6, x_5)$

$\lambda_5^{(3)} - \lambda_3^{(3)} = 15 > l(x_3, x_5)$; $\lambda_5^{(4)} = 20$

$\lambda_5^{(3)} - \lambda_2^{(3)} = 17 > l(x_2, x_5)$

$\lambda_4^{(3)} - \lambda_2^{(3)} = 9 > l(x_2, x_4)$; $\lambda_4^{(4)} = 12$

$\lambda_4^{(3)} - \lambda_3^{(3)} = 7 = l(x_3, x_4)$

$\lambda_3^{(3)} - \lambda_0^{(3)} = 5 = l(x_0, x_3)$; $\lambda_3^{(4)} = 5$

$\lambda_2^{(3)} - \lambda_0^{(3)} = 3 = l(x_0, x_2)$; $\lambda_2^{(4)} = 3$

$\lambda_1^{(3)} - \lambda_2^{(3)} = 6 = l(x_2, x_1)$; $\lambda_1^{(4)} = 9$

$\lambda_1^{(3)} - \lambda_0^{(3)} = 9 > l(x_0, x_1)$; $\lambda_0^{(4)} = 0$

În etapa IV obținem diferențele:

$\lambda_{13}^{(4)} - \lambda_{12}^{(4)} = 15 > l(x_{12}, x_{13})$

$\lambda_{13}^{(4)} - \lambda_{11}^{(4)} = 16 > l(x_{11}, x_{13})$;

$\lambda_{13}^{(4)} - \lambda_{10}^{(4)} = 10 < l(x_{10}, x_{13}) = 11$; $\lambda_{13}^{(5)} = \lambda_{10}^{(4)} + l(x_{10}, x_{13}) = 37 + 11 = 48$

$\lambda_{12}^{(4)} - \lambda_{11}^{(4)} = 1 < l(x_{11}, x_{12}) = 5$; $\lambda_{12}^{(5)} = \lambda_{11}^{(4)} + l(x_{11}, x_{12}) = 31 + 5 = 36$

$\lambda_{12}^{(4)} - \lambda_5^{(4)} = 12\ l(x_5, x_{12})$

$\lambda_{12}^{(4)} - \lambda_7^{(4)} = 6 = l(x_7, x_{12})$

Scanned with OKEN Scanner

$\lambda_{11}^{(4)} - \lambda_{8}^{(4)} = 5 = l(x_8, x_{11})$; $\lambda_{11}^{(5)} = 31$
$\lambda_{11}^{(4)} - \lambda_{5}^{(4)} = 11 > l(x_5, x_{11})$

$\lambda_{10}^{(4)} - \lambda_{11}^{(4)} = 6 < l(x_{11}, x_{10}) = 10$; $\lambda_{10}^{(5)} = \lambda_{11}^{(4)} + l(x_{11}, x_{10}) = 31 + 10 = 41$
$\lambda_{10}^{(4)} - \lambda_{9}^{(4)} = 17 > l(x_9, x_{10})$
$\lambda_{10}^{(4)} - \lambda_{8}^{(4)} = 11 > l(x_8, x_{10})$

$\lambda_{9}^{(4)} - \lambda_{4}^{(4)} = 8 = l(x_4, x_9)$; $\lambda_{9}^{(5)} = 20$

$\lambda_{8}^{(4)} - \lambda_{5}^{(4)} = 6 = l(x_5, x_8)$
$\lambda_{8}^{(4)} - \lambda_{4}^{(4)} = 14 > l(x_4, x_8)$ $\lambda_{8}^{(5)} = 26$

$\lambda_{7}^{(4)} - \lambda_{6}^{(4)} = 11 > l(x_6, x_7)$
$\lambda_{7}^{(4)} - \lambda_{5}^{(4)} = 6 = l(x_5, x_7)$ $\lambda_{7}^{(5)} = 26$

$\lambda_{6}^{(4)} - \lambda_{2}^{(4)} = 12 > l(x_2, x_6)$
$\lambda_{6}^{(4)} - \lambda_{1}^{(4)} = 6 = l(x_1, x_6)$ $\lambda_{6}^{(5)} = 15$

$\lambda_{5}^{(4)} - \lambda_{6}^{(4)} = 5 = l(x_6, x_5)$
$\lambda_{5}^{(4)} - \lambda_{3}^{(4)} = 15 = l(x_3, x_5)$; $\lambda_{5}^{(5)} = 20$
$\lambda_{5}^{(4)} - \lambda_{2}^{(4)} = 17 > l(x_2, x_5)$

$\lambda_{4}^{(4)} - \lambda_{2}^{(4)} = 9 > l(x_2, x_4)$
$\lambda_{4}^{(4)} - \lambda_{3}^{(4)} = 7 = l(x_3, x_4)$ $\lambda_{4}^{(5)} = 12$

$\lambda_{3}^{(4)} - \lambda_{0}^{(4)} = 5 = l(x_0, x_3)$ $\lambda_{3}^{(5)} = 5$

$\lambda_{2}^{(4)} - \lambda_{0}^{(4)} = 3 = l(x_0, x_2)$ $\lambda_{2}^{(5)} = 3$

$\lambda_{1}^{(4)} - \lambda_{2}^{(4)} = 6 = l(x_2, x_1)$ $\lambda_{1}^{(5)} = 9$

$\lambda_{1}^{(4)} - \lambda_{0}^{(4)} = 9 > l(x_0, x_1)$ $\lambda_{0}^{(5)} = 0$

Calculăm diferențele corespunzătoare acestei etape:

$\lambda_{13}^{(5)} - \lambda_{12}^{(5)} = 12 > l(x_{12}, x_{13})$
$\lambda_{13}^{(5)} - \lambda_{11}^{(5)} = 17 > l(x_{11}, x_{13})$
$\lambda_{13}^{(5)} - \lambda_{10}^{(5)} = 7 < l(x_{10}, x_{13}) = 11$; $\lambda_{13}^{(6)} = \lambda_{10}^{(5)} + l(x_{10}, x_{13}) = 41 + 11 = 52$

$\lambda_{12}^{(5)} - \lambda_{11}^{(5)} = 5 = l(x_{11}, x_{12})$
$\lambda_{12}^{(5)} - \lambda_{5}^{(5)} = 16 = l(x_3, x_{12})$ $\lambda_{12}^{(6)} = 36$
$\lambda_{12}^{(5)} - \lambda_{7}^{(5)} = 10 > l(x_7, x_{12})$

Scanned with OKEN Scanner

$\lambda_{11}^{(5)} - \lambda_8^{(5)} = 5 = l(x_8, x_{11})$ $\quad \lambda_{11}^{(6)} = 31$
$\lambda_{11}^{(5)} - \lambda_5^{(5)} = 11 > l(x_5, x_{11})$

$\lambda_{10}^{(5)} - \lambda_{11}^{(5)} = 10 = l(x_{11}, x_{10})$
$\lambda_{10}^{(5)} - \lambda_9^{(5)} = 21 > l(x_9, x_{10})$ $\quad \lambda_{10}^{(6)} = 41$
$\lambda_{10}^{(5)} - \lambda_8^{(5)} = 15 > l(x_8, x_{10})$

$\lambda_9^{(5)} - \lambda_4^{(5)} = 8 = l(x_4, x_9)$ $\quad \lambda_9^{(6)} = 20$

$\lambda_8^{(5)} - \lambda_5^{(5)} = 6 = l(x_5, x_8)$ $\quad \lambda_8^{(6)} = 26$
$\lambda_8^{(5)} - \lambda_4^{(5)} = 14 > l(x_4, x_8)$

$\lambda_7^{(5)} - \lambda_6^{(5)} = 11 > l(x_6, x_7)$ $\quad \lambda_7^{(5)} = 26$
$\lambda_7^{(5)} - \lambda_5^{(5)} = 6 = l(x_5, x_7)$

$\lambda_6^{(5)} - \lambda_2^{(5)} = 12 > l(x_2, x_6)$ $\quad \lambda_6^{(6)} = 15$
$\lambda_6^{(5)} - \lambda_1^{(5)} = 6 = l(x_1, x_6)$

$\lambda_5^{(5)} - \lambda_6^{(5)} = 5 = l(x_6, x_5)$
$\lambda_5^{(5)} - \lambda_3^{(5)} = 15 > l(x_3, x_5)$; $\quad \lambda_5^{(6)} = 20$
$\lambda_5^{(5)} - \lambda_2^{(5)} = 17 > l(x_2, x_5)$

$\lambda_4^{(5)} - \lambda_2^{(5)} = 9 > l(x_2, x_4)$ $\quad \lambda_4^{(6)} = 12$
$\lambda_4^{(5)} - \lambda_3^{(5)} = 7 = l(x_3, x_4)$

$\lambda_3^{(5)} - \lambda_0^{(5)} = 5 = l(x_0, x_3)$; $\quad \lambda_3^{(6)} = 5$

$\lambda_2^{(5)} - \lambda_0^{(5)} = 3 = l(x_0, x_2)$ $\quad \lambda_2^{(6)} = 3$

$\lambda_1^{(5)} - \lambda_2^{(5)} = 6 = l(x_2, x_1)$ $\quad \lambda_1^{(6)} = 9$
$\lambda_1^{(5)} - \lambda_0^{(5)} = 9 > l(x_0, x_1)$

Deoarece diferența $\lambda_{13} - \lambda_{10} < l(x_{10}, x_{13})$, se impune corectarea lui λ_{13} și recalcularea diferențelor pentru această etapă. Astfel, avem:

$\lambda_{13}^{(6)} - \lambda_{12}^{(6)} = 16 > l(x_{12}, x_{13})$
$\lambda_{12}^{(6)} - \lambda_{11}^{(6)} = 21 > l(x_{11}, x_{13})$
$\lambda_{13}^{(6)} - \lambda_{10}^{(6)} = 11 = l(x_{10}, x_{13})$

$\lambda_{12}^{(6)} - \lambda_{11}^{(6)} = 5 = l(x_{11}, x_{12})$
$\lambda_{12}^{(6)} - \lambda_5^{(6)} = 16 > l(x_5, x_{12})$
$\lambda_{12}^{(6)} - \lambda_7^{(6)} = 10 > l(x_7, x_{12})$

Scanned with OKEN Scanner

$\lambda_{11}^{(5)} - \lambda_{8}^{(5)} = 5 = l(x_8, x_{11})$
$\lambda_{11}^{(5)} - \lambda_{5}^{(5)} = 11 > l(x_5, x_{11})$ $\quad \lambda_{11}^{(6)} = 31$

$\lambda_{10}^{(5)} - \lambda_{11}^{(5)} = 10 = l(x_{11}, x_{10})$
$\lambda_{10}^{(5)} - \lambda_{9}^{(5)} = 21 > l(x_9, x_{10})$ $\quad \lambda_{10}^{(6)} = 41$
$\lambda_{10}^{(5)} - \lambda_{8}^{(5)} = 15 > l(x_8, x_{10})$

$\lambda_{9}^{(5)} - \lambda_{4}^{(5)} = 8 = l(x_4, x_9)$ $\quad \lambda_{9}^{(6)} = 20$

$\lambda_{8}^{(5)} - \lambda_{5}^{(5)} = 6 = l(x_5, x_8)$ $\quad \lambda_{8}^{(6)} = 26$
$\lambda_{8}^{(5)} - \lambda_{4}^{(5)} = 14 > l(x_4, x_8)$

$\lambda_{7}^{(5)} - \lambda_{6}^{(5)} = 11 > l(x_6, x_7)$ $\quad \lambda_{7}^{(5)} = 26$
$\lambda_{7}^{(5)} - \lambda_{5}^{(5)} = 6 = l(x_5, x_7)$

$\lambda_{6}^{(5)} - \lambda_{2}^{(5)} = 12 > l(x_2, x_6)$ $\quad \lambda_{6}^{(6)} = 15$
$\lambda_{6}^{(5)} - \lambda_{1}^{(5)} = 6 = l(x_1, x_6)$

$\lambda_{5}^{(5)} - \lambda_{6}^{(5)} = 5 = l(x_6, x_5)$
$\lambda_{5}^{(5)} - \lambda_{3}^{(5)} = 15 > l(x_3, x_5)$; $\quad \lambda_{5}^{(6)} = 20$
$\lambda_{5}^{(5)} - \lambda_{2}^{(5)} = 17 > l(x_2, x_5)$

$\lambda_{4}^{(5)} - \lambda_{2}^{(5)} = 9 > l(x_2, x_4)$ $\quad \lambda_{4}^{(6)} = 12$
$\lambda_{4}^{(5)} - \lambda_{3}^{(5)} = 7 = l(x_3, x_4)$

$\lambda_{3}^{(5)} - \lambda_{0}^{(5)} = 5 = l(x_0, x_3)$; $\quad \lambda_{3}^{(6)} = 5$

$\lambda_{2}^{(5)} - \lambda_{0}^{(5)} = 3 = l(x_0, x_2)$ $\quad \lambda_{2}^{(6)} = 3$

$\lambda_{1}^{(5)} - \lambda_{2}^{)5)} = 6 = l(x_2, x_1)$ $\quad \lambda_{1}^{(6)} = 9$
$\lambda_{1}^{(5)} - \lambda_{0}^{(5)} = 9 > l(x_0, x_1)$

Deoarece diferența $\lambda_{13} - \lambda_{10} < l(x_{10}, x_{13})$, se impune corectarea lui λ_{13} și recalcularea diferențelor pentru această etapă. Astfel, avem:

$\lambda_{13}^{(6)} - \lambda_{12}^{(6)} = 16 > l(x_{12}, x_{13})$
$\lambda_{12}^{(6)} - \lambda_{11}^{(6)} = 21 > l(x_{11}, x_{13})$
$\lambda_{13}^{(6)} - \lambda_{10}^{(6)} = 11 = l(x_{10}, x_{13})$

$\lambda_{12}^{(6)} - \lambda_{11}^{(6)} = 5 = l(x_{11}, x_{12})$
$\lambda_{12}^{(6)} - \lambda_{5}^{(6)} = 16 > l(x_5, x_{12})$
$\lambda_{12}^{(6)} - \lambda_{7}^{(6)} = 10 > l(x_7, x_{12})$

Scanned with OKEN Scanner

$\lambda_{11}^{(6)} - \lambda_8^{(6)} = 5 = l(x_8, x_{11})$
$\lambda_{11}^{(6)} - \lambda_5^{(6)} = 11 > l(x_5, x_{11})$

$\lambda_{10}^{(6)} - \lambda_{11}^{(6)} = 10 = l(x_{11}, x_{10})$
$\lambda_{10}^{(6)} - \lambda_9^{(6)} = 21 > l(x_9, x_{10})$
$\lambda_{10}^{(6)} - \lambda_8^{(6)} = 15 > l(x_8, x_{10})$

$\lambda_9^{(6)} - \lambda_4^{(6)} = 8 = l(x_4, x_6)$

$\lambda_8^{(6)} - \lambda_5^{(6)} = 6 = l(x_5, x_8)$
$\lambda_8^{(6)} - \lambda_4^{(6)} = 14 > l(x_4, x_8)$

$\lambda_7^{(6)} - \lambda_6^{(6)} = 11 > l(x_6, x_7)$
$\lambda_7^{(6)} - \lambda_5^{(6)} = 6 = l(x_5, x_7)$

$\lambda_6^{(6)} - \lambda_2^{(6)} = 12 > l(x_2, x_6)$
$\lambda_6^{(6)} - \lambda_1^{(6)} = 6 = l(x_1, x_6)$

$\lambda_5^{(6)} - \lambda_6^{(6)} = 5 = l(x_6, x_5)$
$\lambda_5^{(6)} - \lambda_3^{(6)} = 15 > l(x_3, x_5)$
$\lambda_5^{(6)} - \lambda_2^{(6)} = 17 > l(x_2, x_5)$

$\lambda_4^{(6)} - \lambda_2^{(6)} = 9 > l(x_2, x_4)$
$\lambda_4^{(6)} - \lambda_3^{(6)} = 7 = l(x_3, x_4)$

$\lambda_3^{(6)} - \lambda_0^{(6)} = 5 = l(x_0, x_3)$

$\lambda_2^{(6)} - \lambda_0^{(6)} = 3 = l(x_0, x_2)$

$\lambda_1^{(6)} - \lambda_2^{(6)} = 6 = l(x_2, x_1)$
$\lambda_1^{(6)} - \lambda_0^{(6)} = 9 > l(x_0, x_1)$

Întrucît în această ultimă etapă am obținut toate diferențele

$$\lambda_j^{(6)} - \lambda_i^{(6)} \geqslant l(x_i, x_j),$$

algoritmul indică sistarea operațiilor.

Lungimea maximă a drumului este

$$\lambda_{13}^{(6)} = 52,$$

Scanned with OKEN Scanner

iar mulţimea de vîrfuri corespunzătoare acestui drum este

$$\{x_0, x_2, x_1, x_6, x_5, x_8, x_{10}, x_{11}, x_{13}\}.$$

Pentru determinarea drumului maxim au fost luate în consideraţie următoarele diferenţe:

$$\lambda_{13}^{(6)} - \lambda_{10}^{(6)} = 11 = l(x_{10}, x_{13})$$
$$\lambda_{10}^{(6)} - \lambda_{11}^{(6)} = 10 = l(x_{11}, x_{10})$$
$$\lambda_{11}^{(6)} - \lambda_{8}^{(6)} = 5 = l(x_8, x_{11})$$
$$\lambda_{8}^{(6)} - \lambda_{5}^{(6)} = 6 = l(x_5, x_8)$$
$$\lambda_{5}^{(6)} - \lambda_{6}^{(6)} = 5 = l(x_6, x_5)$$
$$\lambda_{6}^{(6)} - \lambda_{1}^{(6)} = 6 = l(x_2, x_1)$$
$$\lambda_{2}^{(6)} - \lambda_{0}^{(6)} = 3 = l(x_0, x_2)$$

Însumînd membru cu membru egalităţile, obţinem:

$$\lambda_{13}^{(6)} - \lambda_{0}^{(6)} = 11 + 10 + 5 + 6 + 5 + 6 + 6 + 3 =$$
$$= l(x_{10}, x_{13}) + l(x_{11}, x_{10}) + l(x_8, x_{11}) + l(x_5, x_8) +$$
$$+ l(x_6, x_5) + l(x_1, x_6) + l(x_2, x_1) + l(x_0, x_2) = 52$$

De unde, ţinînd seama că $\lambda_0^{(6)} = 0$, rezultă $\lambda_{13}^{(6)} = 52 = l(d)$

2°. Algoritmul Bellman-Kalaba pentru găsirea drumului de lungime maximă. Acest algoritm este o variantă a celui prezentat cu ocazia determinării drumului de lungime minimă şi care se bazează pe aceeaşi teoremă, din programarea dinamică a lui Bellman, care afirmă că drumul de lungime maximă se compune din drumuri parţiale maxime.

Şi de această dată în prima etapă se asociază grafului $G = (X, F)$, finit şi fără circuite, o matrice ale cărei elemente sînt:

$$r_{ij} = \begin{cases} l(x_i \cdot x_j) = G_{ij}, & \text{dacă există arcul } (x_i, x_j), \\ 0 & \text{dacă } i = j, \\ -\infty & \text{dacă nu există arc de la } x_i \text{ la } x_j. \end{cases}$$

În vederea găsirii drumului

$$d = (x_0, x_{i_1}, x_{i_2}, \ldots x_{i_p} \ldots x_n),$$

pentru care

$$l(x_0, x_{i_1}) + l(x_{i_1}, x_{i_2}) + \ldots + l(x_{i_{p-1}}) x_{i_p}) + \ldots + l(x_{i_q}, x_n)$$

să fie maximă, se procedează în etape.

Etapa I. Pentru determinarea acestui drum, se consideră:

$$V_i^{(0)} = C_{in} \qquad i = 0,1,2, \ldots, n-1$$

$$V_n^{(0)} = 0$$

Etapa a II-a. Calculăm

$$V_i^{(1)} = \max_{\substack{j \neq i \\ j = 1,2,\ldots,n}} \{V_j^{(0)} + C_{ij}\} \; i = 0,1,2, \ldots, n-1,$$

$$V_n^{(1)} = 0;$$

$$V_i^{(2)} = \max_{\substack{j \neq i \\ j = 1,2,\ldots,n}} \{V_j^{(1)} + C_{ij}\} \quad i = 0,1,2, \ldots, n-1,$$

$$V_n^{(2)} = 0;$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$V_i^{(k)} = \max_{\substack{j \neq i \\ j = 1,2,\ldots,n}} \{V_j^{k-1} + C_{ij}\} \; i = 0,1,2, \ldots m-1,$$

$$V_n^{(k)} = 0;$$

$$V_i^{(k+1)} = \max_{\substack{j \neq i \\ j = 0,1,\ldots,n}} \{V_j^{(k)} + C_{ij}\} \; i = 0,1,2, \ldots, n-1,$$

$$V_n^{(k)} = 0.$$

Etapa a III-a. Dacă coloanele $V_j^{(k)}$ și $V_j^{(k+1)}$ coincid, adică $V_j^{(k)} = V_j^{(k+1)}$ pentru orice $i = 0,1,2, \ldots, n$, algoritmul ia sfîrșit, găsindu-se drumul de lungime maximă.

Lungimea maximă a drumului este de:

$$\max\, l(d) = V_0^{(k)} = V_0^{(k+1)}.$$

Să aplicăm algoritmul Bellman-Kalaba, pentru găsirea drumului de lungime maximă între vîrfurile x_0 și x_{13}, în graful din figura II.3.1.

Matricea asociată acestui graf este dată de tabelul T.

Scanned with OKEN Scanner

Tabelul T

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	$V_i^{(0)}$	$V_i^{(1)}$	$V_i^{(2)}$	$V_i^{(3)}$	$V_i^{(4)}$	$V_i^{(5)}$	$V_i^{(6)}$	$V_i^{(7)}$	$V_i^{(8)}$
x_0	0	1	3	5	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	33	43	47	48	52	52
x_1	$-\infty$	0	$-\infty$	$-\infty$	$-\infty$	$-\infty$	6	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	29	39	43	43	43	43
x_2	$-\infty$	6	0	$-\infty$	6	7	8	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	23	35	41	45	49	49	49
x_3	$-\infty$	$-\infty$	$-\infty$	0	7	10	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	28	38	42	42	42	42	42
x_4	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	$-\infty$	$-\infty$	$-\infty$	9	8	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	28	35	35	35	35	35	35
x_5	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	$-\infty$	6	6	$-\infty$	$-\infty$	7	8	$-\infty$	$-\infty$	16	28	32	32	32	32	32	32
x_6	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	5	0	7	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	23	33	37	37	37	37	37
x_7	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	$-\infty$	$-\infty$	$-\infty$	$-\infty$	6	$-\infty$	$-\infty$	14	14	14	14	14	14	14	14
x_8	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	$-\infty$	7	5	$-\infty$	$-\infty$	$-\infty$	18	26	26	26	26	26	26	26
x_9	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	9	$-\infty$	$-\infty$	$-\infty$	$-\infty$	20	20	20	20	20	20	20	20
x_{10}	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	$-\infty$	$-\infty$	11	11	11	11	11	11	11	11	11	11
x_{11}	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	10	0	5	6	6	21	21	21	21	21	21	21	21
x_{12}	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	8	8	8	8	8	8	8	8	8	8
x_{13}	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	-8	$-\infty$	$-\infty$	$-\infty$	0	0	0	0	0	0	0	0	0	0

Scanned with OKEN Scanner

Matricea T este completată de coloana $V_i^{(0)}$, care coincide cu ultima sa coloană.

Pentru calculul celorlalte coloane $V_i^{(h)}$, s-au făcut calculele indicate de algoritm, și anume:

$V_0^{(1)} = \max \{ 1 - \infty ; \ 3 - \infty ; \ 5 - \infty ; - \infty - \infty ; - - \infty - \infty ; - \infty - \infty ; - \infty - \infty ; - \infty - \infty ; - \infty - - \infty ; \ - \infty - \infty ; \ - \infty + 11 + - \infty + 6 ; \ - \infty + 8 ; - \infty + 0 \} = - \infty .$

$V_1^{(1)} = \max \{ - \infty - \infty ; - \infty - \infty ; - \infty - \infty ; - \infty - - \infty ; - \infty - \infty ; 6 - \infty ; - \infty - \infty ; - \infty - \infty ; - \infty + + 11 ; - \infty + 6 ; - \infty + 8 ; - \infty + 0 \} = - \infty .$

$V_2^{(1)} = \max \{ - \infty - \infty , 6 - \infty , - \infty - \infty , 6 - \infty , 7 - - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty + 11 , - - \infty + 6 , - \infty + 8 , - \infty + 0 \} = - \infty .$

$V_3^{(1)} = \max \{ - \infty - \infty , - \infty - \infty , - \infty - \infty , 7 - \infty , 10 - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty + + 11 , - \infty + 6 , - \infty + 8 , - \infty + 0 \} = - \infty$

$V_4^{(1)} = \max \{ - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , 9 - \infty , 8 - \infty , - \infty + 11 , - \infty + 6 , - \infty + 0 \} = - \infty .$

$V_5^{(1)} = \max \{ - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , 6 - \infty , 6 - \infty , - \infty - \infty , - \infty + 11 , 7 + 6 , 8 + 8 , - \infty + 0 \} = 16.$

$V_6^{(1)} = \max \{ - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , 5 - \infty , 7 - \infty , - \infty - \infty , - \infty - \infty , \infty + 11 , - \infty + 6 , - \infty - \infty , - \infty + 0 \} = - \infty .$

$V_7^{(1)} = \max \{ - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - - \infty , \infty + 11 , - \infty + 6 , 6 + 8 , - \infty + 0 \} = 14.$

$V_8^{(1)} = \max \{ - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - - \infty , 7 + 11 , 5 + 6 , - \infty + 8 , - \infty + 0 \} = 18.$

$V_9^{(1)} = \max \{ - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , - \infty - \infty , 9 + 11 , - \infty + 6 , - \infty + 8 , - \infty + 0 \} = 20.$

Scanned with OKEN Scanner

$V_{10}^{(1)} = \max \{-\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty + 6, -\infty + 8, 11 + 0\} = 11.$

$V_{11}^{(1)} = \max \{-\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, 10 + 11, 5 + 8, 6 + 10\} = 21.$

$V_{12}^{(1)} = \max \{-\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty - \infty, -\infty + 11, -\infty + 6, 8 + 0\} = 8.$

$V_{13}^{(1)} = 0.$

$V_i^{(1)}$ au fost trecute în coloana din prelungirea matricei. La fel au fost calculate și celelalte valori $V_i^{(k)}$ $i = 0,1,2,\ldots, 13$ pentru $k = 2, 3, 4, 5, 6, 7, 8.$

Se constată că:

$$V_i^{(7)} = V_i^{(7)} \; i = 0,1,2, \ldots, 13,$$

ceea ce înseamnă că $V_0^{(7)} = V_0^{(8)} = 52$ este valoarea drumului de lungime maximă în acest graf, drum determinat pe baza algoritmului Bellman-Kalaba.

§ 5. DRUMURI HAMILTONIENE ÎNTR-UN GRAF

În capitolul I s-a arătat că drumul hamiltonian este acela care trece prin toate vîrfurile grafului, folosind fiecare vîrf o singură dată.

Căutarea unor asemenea drumuri este legată de problemele ridicate de planificare, de stabilirea ordinei de lansare în producție a unor repere, de asamblarea unor lucrări complexe, precum și altele.

Asemenea activități cuprind în general mai multe operații (faze) a căror ordine trebuie stabilită în așa fel, încît executarea ansamblului de operații să fie cea mai avantajoasă din punct de vedere economic.

Pentru înțelegerea celor de mai sus, să presupunem că pentru executarea unui produs de largă circulație sînt necesare cinci operații $(x_1, x_2, x_3, x_4, x_5)$. Ele trebuie să se succeadă în următoarea ordine:

Scanned with OKEN Scanner

— operația x_1 trebuie să fie urmată de operațiile x_2 și x_3;
— operația x_2 trebuie să fie urmată de x_3 și de x_5;
— operația x_3 trebuie să fie urmată de x_4;
— operația x_4 trebuie să fie urmată de x_2;
— operația x_5 trebuie să fie urmată de x_4.

Această condiționare a operațiilor conduce la gradul din figura II.5.1.

Analizînd acest graf, remarcăm existența următoarelor drumuri hamiltoniene:

$$d_{H_1} = (x_1, x_2, x_5, x_3, x_4),$$
$$d_{H_2} = (x_1, x_3, x_4, x_2, x_5),$$

care ne dau posibilitatea alegerii succesiunii operațiilor în două moduri distincte, cu condiția ca toate operațiile să fie executate o dată și numai o dată și să fie respectată în același timp ordinea stabilită de execuție.

În exemplul dat, s-a presupus executarea unui număr mic de operații, ceea ce ne-a dat posibilitatea să determinăm drumul hamiltonian (ordinea de lansare) prin simpla urmărire a grafului.

Dacă numărul operațiilor este mare, simpla cercetare a grafului este cu totul insuficientă și uneori chiar practic imposibilă. Din această cauză, s-au elaborat metode speciale prin care să se determine drumurile hamiltoniene.

Dintre acestea, descriem mai întîi metoda cunoscută sub numele de „algoritmul lui Foulkes“, care prezintă avantajul

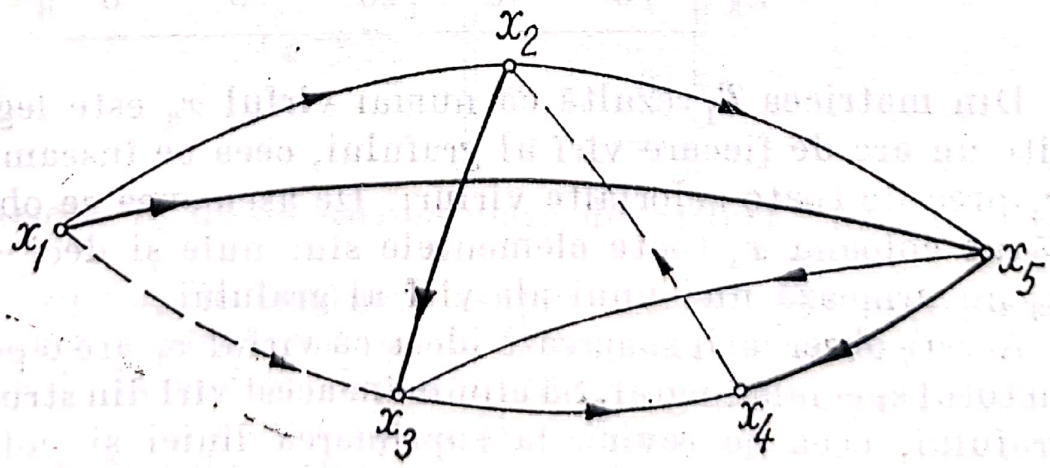


Fig. II. 5. 1

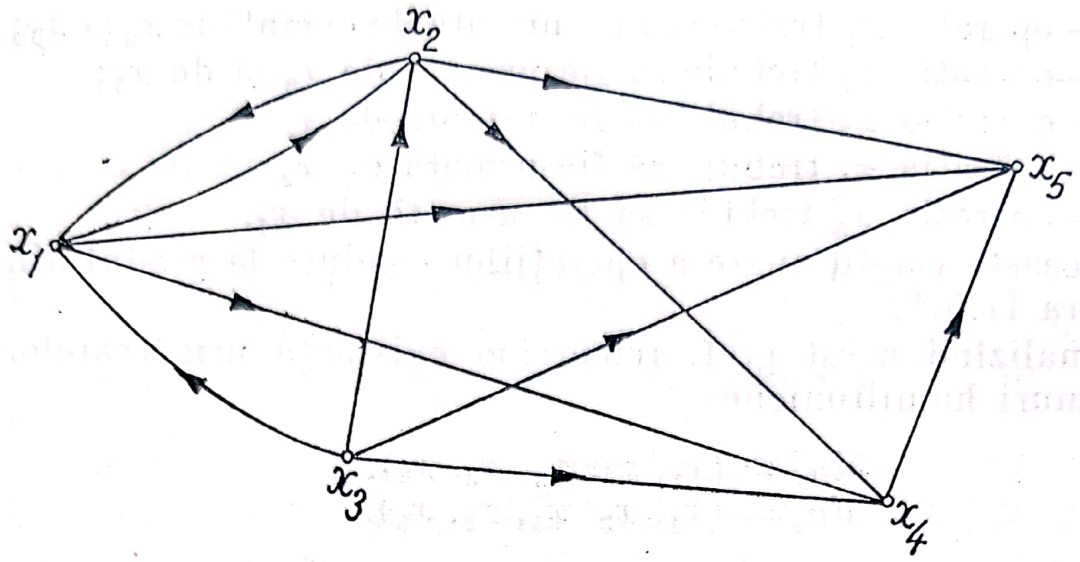


Fig. II. 5. 2

că descompune graful cu multe vîrfuri în subgrafe tare conexe, al căror studiu este mult mai comod în aplicații.

Vom prezenta algoritmul pornind de la graful din figura II.5.2.

Acestui graf îi corespunde matricea booleană din tabelul T_1:

$$T_1 = \begin{array}{c|ccccc} & x_1 & x_2 & x_3 & x_4 & x_5 \\ \hline x_1 & 0 & 1 & 0 & 1 & 1 \\ x_2 & 1 & 0 & 0 & 1 & 1 \\ x_3 & 1 & 1 & 0 & 1 & 1 \\ x_4 & 0 & 0 & 0 & 0 & 1 \\ x_5 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Din matricea T_1 rezultă că numai vîrful x_3 este legat cu cîte un arc de fiecare vîrf al grafului, ceea ce înseamnă că x_3 precede toate celorlalte vîrfuri. De asemenea se observă că pe coloana x_3 toate elementele sînt nule și deci vîrful x_3 nu urmează nici unui alt vîrf al grafului.

Aceste observații sugerează ideea că vîrful x_3 are o poziție cu totul specială în graf. Să eliminăm acest vîrf din structura grafului, ceea ce revine la suprimarea liniei și coloanei corespunzătoare vîrfului x_3 din matricea T_1.

Scanned with OKEN Scanner

Obținem în acest fel matricea T_2 care va conține mai puțin linia și coloana x_3:

$$T_2 = \begin{array}{c|cccc} & x_1 & x_2 & x_4 & x_5 \\ \hline x_1 & 0 & 1 & 1 & 1 \\ x_2 & 1 & 0 & 1 & 1 \\ x_4 & 0 & 0 & 0 & 1 \\ x_5 & 0 & 0 & 0 & 0 \end{array}$$

În matricea T_2, suprimăm linia și coloana x_5, deoarece vîrful x_5 este legat de toate celelalte vîrfuri, respectiv x_1, x_2, x_4 prin orice incidente spre interior vîrfului x_5. Linia și coloana x_5 se suprimă, obținînd tabela T_3:

$$T_3 = \begin{array}{c|ccc} & x_1 & x_2 & x_4 \\ \hline x_1 & 0 & 1 & 1 \\ x_2 & 1 & 0 & 1 \\ x_4 & 0 & 0 & 0 \end{array}$$

Din aceleași motive, în T_3 suprimăm linia și coloana x_4, obținîndu-se tabela T_4:

$$T_4 = \begin{array}{c|cc} & x_1 & x_2 \\ \hline x_1 & 0 & 1 \\ x_2 & 1 & 0 \end{array}$$

Matricea T_4 ne dă ultima clasă de vîrfuri, formată din (x_1, x_2).

Prin operațiile efectuate asupra acestui graf obținem o clasificare a vîrfurilor în clase ordonate și anume:

clasa I de vîrfuri, formată din x_3, obținută prin suprimarea liniei și coloanei x_3;

clasa a II-a de vîrfuri, formată din x_5, obținută prin suprimarea coloanei și liniei x_5;

Scanned with OKEN Scanner

clasa a III-a de vîrfuri, formată din x_4, găsită o dată cu suprimarea coloanei și liniei x_4;

clasa a IV-a de vîrfuri, formată din x_1, x_2, ce se obține prin suprimarea liniilor și coloanelor x_1, x_2.

Ținînd seama de această ordonare a vîrfurilor, în clase de echivalență, obținem graful ordonat din figura II.5.3.

Pe figura II.5.3 putem citi cu ușurință drumurile hamiltoniene, ținînd seama de clasele de echivalență obținute.

Astfel, avem drumurile hamiltoniene:

$$d_{H_1} = \{x_3,\ x_3,\ x_1,\ x_4,\ x_5\}$$
$$d_{H_2} = \{x_3,\ x_1,\ x_2,\ x_4,\ x_5\}.$$

Găsirea drumurilor hamiltoniene prin procedeul descris a fost posibilă datorită faptului că atît în matricea inițială T_1 cît și în matricele T_2 și T_3 vîrfurile eliminate aveau o poziție cu totul particulară și anume:

a) coloana corespunzătoare vîrfului (vîrfurilor) eliminat (e) era formată numai din elemente egale cu zero, iar linia (liniile) corespunzătoare — numai din elemente egale cu unitatea cu excepția elementului de la diagonală care este zero;

b) coloana corespunzătoare vîrfului (vîrfurilor) eliminat formată numai din elemente egale cu unitatea, iar linia (liniile) corespunzătoare formate numai din elemente egale cu zero, cu excepția elementului de pe diagonală care este zero.

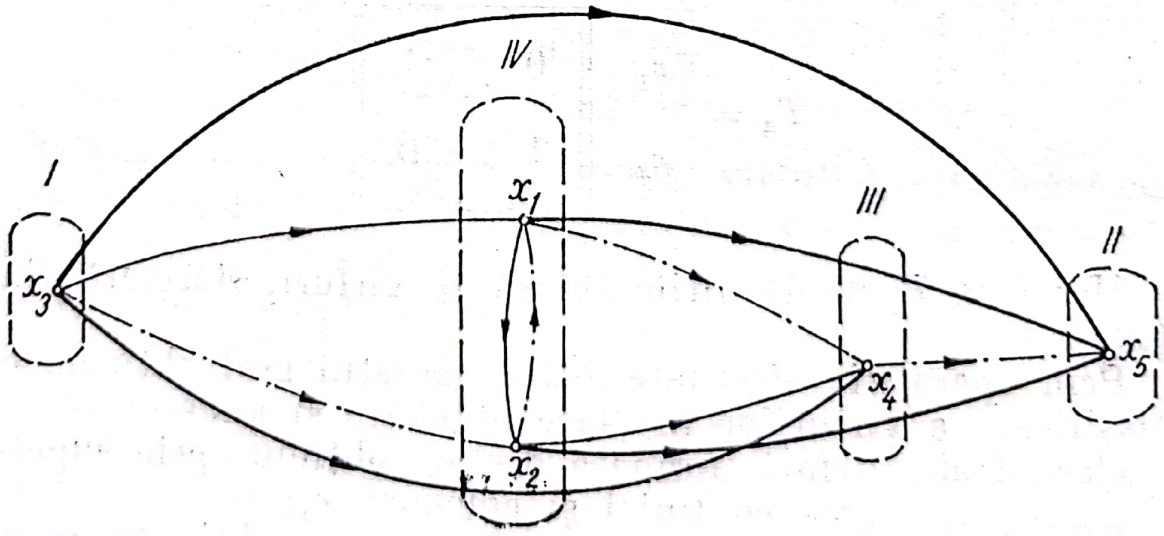


Fig. II.5.3

Scanned with OKEN Scanner

În cazul (a) vîrful eliminat precede toate celelalte vîrfuri ale grafului, iar în cazul (b) vîrful eliminat este urmaşul tuturor vîrfurilor.

Tocmai acest lucru a permis posibilitatea trecerii de la o etapă la alta, ceea ce nu se întîmplă însă în toate cazurile întîlnite. Prezentăm în continuare algoritmul lui Foulkes în toată generalitatea lui.

Fie graful conex $G = (X, F)$, ce conţine un număr, n, mare de vîrfuri.

Etapa I. Asociem grafului o matrice de incidenţă a arcelor (matrice booleană), notată ca de obicei cu B şi ale cărei elemente se determină după regulile cunoscute.

Etapa a II-a. Ridicăm la puteri succesive matricea $(E + B)$ unde E este matricea unitate de acelaşi ordin cu matricea B (operaţiile fiind de natură booleană).

Operaţia de ridicare la putere continuă pînă în momentul în care două puteri consecutive ne dau matrice identice, adică:

$$(E + B)^k = (E + B)^{(k+1)}.$$

Etapa a III-a. De fiecare dată cînd prin ridicarea la putere obţinem linii care conţin numai elemente egale cu unu şi coloane corespunzătoare formate numai din elemente egale cu zero, acestea se suprimă.

Continuăm operaţia de suprimare după procedeul indicat, pînă cînd au fost suprimate ultimele linii şi coloane ale matricei.

Cu ocazia fiecărei suprimări obţinem vîrfurile aparţinînd fiecărei clase de echivalenţă.

Dacă am suprimat, de pildă, în prima etapă liniile $(L_{i_1}, L_{i_2}, \ldots, L_{i_p})$, vîrfurile corespunzătoare clasei I de echivalenţă vor fi:

$$x_{i_1}, x_{i_2}, \ldots, x_{i_p}.$$

Analog se procedează şi pentru găsirea celorlalte clase.

Scanned with OKEN Scanner

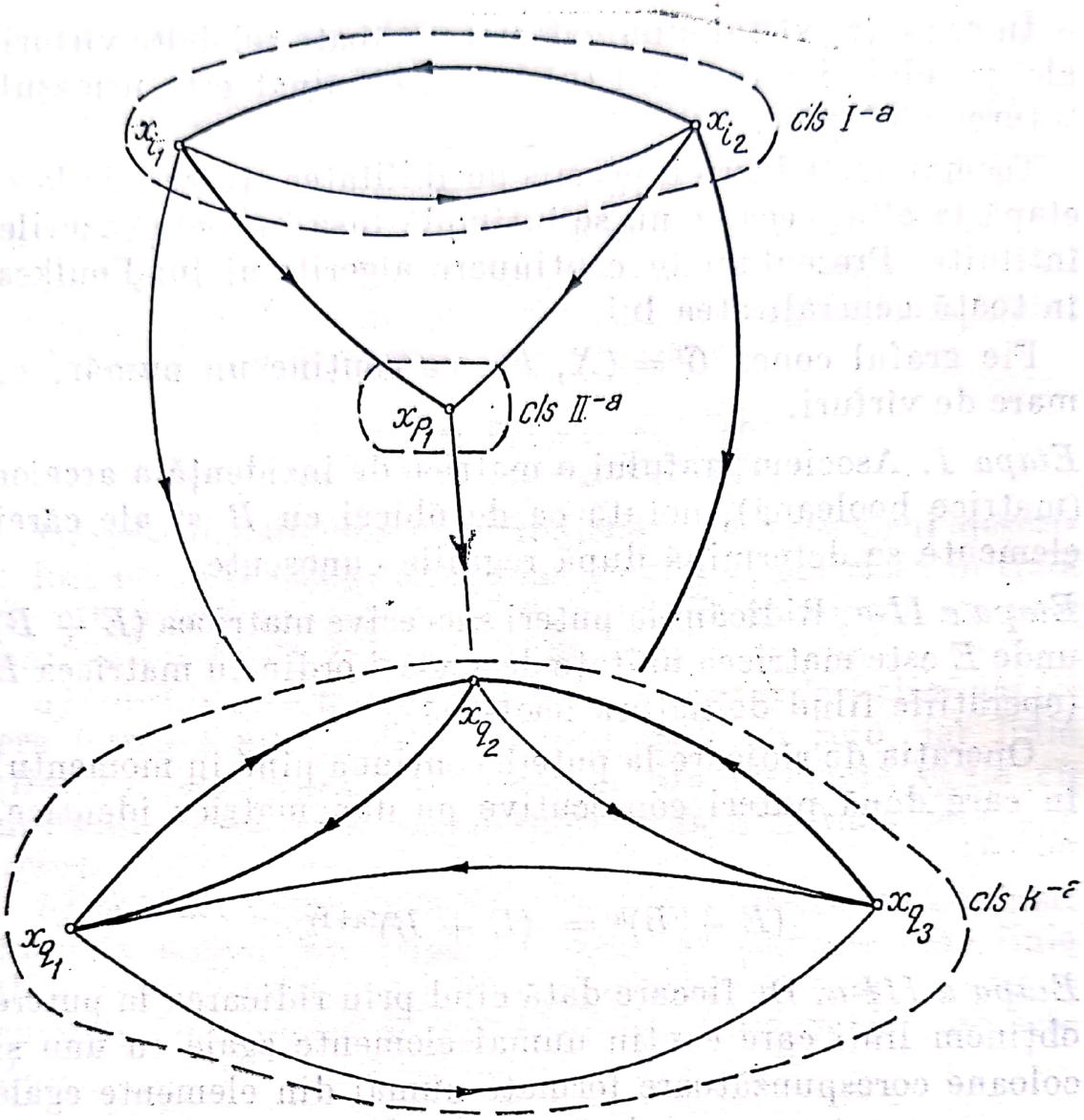


Fig. II.5.4

Etapa a IV-a. După parcurgerea acestor etape, graful inițial va căpăta o nouă înfățișare în funcție de clasele de echivalență obținute, și care se construiește astfel:

a) Reprezentăm mai întîi vîrfurile corespunzătoare primei clase de echivalență, apoi vîrfurile celei de-a doua clase de echivalență și tot așa, pînă la reprezentarea ultimei clase de echivalență.

b) Trasăm arcele corespunzătoare acestor vîrfuri, obținîndu-se în acest fel un graf de forma celui din figura II.5.4.

Pe noul graf se pot citi drumurile hamiltoniene plecînd de la vîrfurile primei clase de echivalență pînă la vîrfurile

Scanned with OKEN Scanner

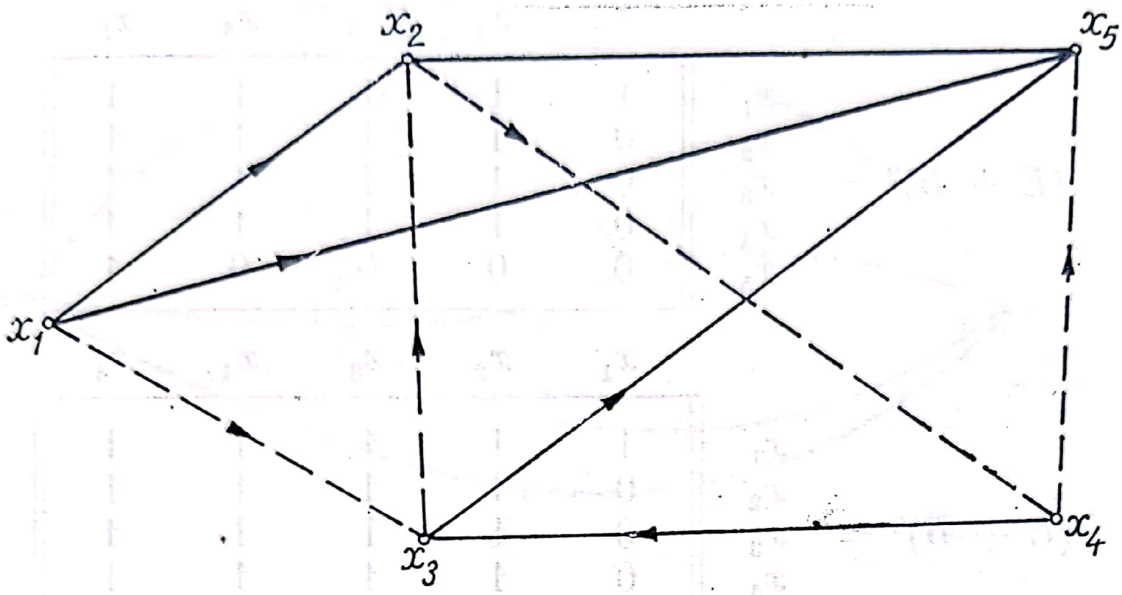


Fig. II. 5.5

ultimei clase, ținînd seama de faptul că fiecare clasă de echivalență constituie un graf tare conex, și de legăturile dintre clase.

Să găsim drumurile hamiltoniene pe graful din figura II.5.5, pe baza celor arătate mai sus:

Etapa I

$B =$

	x_1	x_2	x_3	x_4	x_5
x_1	0	1	1	0	1
x_2	0	0	0	1	1
x_3	0	1	0	0	1
x_4	0	0	1	0	1
x_5	0	0	0	0	0

Etapa a II-a

$(E + B) =$

	x_1	x_2	x_3	x_4	x_5
x_1	1	1	1	0	1
x_2	0	1	0	1	1
x_3	0	1	1	0	1
x_4	0	0	1	1	1
x_5	0	0	0	0	1

$$(E+B)^2 = \begin{array}{c|ccccc} & x_1 & x_2 & x_3 & x_4 & x_5 \\ \hline x_1 & 1 & 1 & 1 & 1 & 1 \\ x_2 & 0 & 1 & 1 & 1 & 1 \\ x_3 & 0 & 1 & 1 & 1 & 1 \\ x_4 & 0 & 1 & 1 & 1 & 1 \\ x_5 & 0 & 0 & 0 & 0 & 1 \end{array}$$

$$(E+B)^3 = \begin{array}{c|ccccc} & x_1 & x_2 & x_3 & x_4 & x_5 \\ \hline x_1 & 1 & 1 & 1 & 1 & 1 \\ x_2 & 0 & 1 & 1 & 1 & 1 \\ x_3 & 0 & 1 & 1 & 1 & 1 \\ x_4 & 0 & 1 & 1 & 1 & 1 \\ x_5 & 0 & 0 & 0 & 0 & 1 \end{array}$$

Deoarece

$$(E+B)^2 = (E+B)^3 ,$$

procedăm la suprimarea liniei și coloanei lui x_1, deci prima clasă de echivalență este formată din vîrful x_1.

În matricea rămasă, $A_1 =$

	x_2	x_3	x_4	x_5
x_2	1	1	1	1
x_3	1	1	1	1
x_4	1	1	1	1
x_5	0	0	0	1

liniile corespunzătoare vîrfurilor x_2, x_3, x_4 au toate elementele egale cu 1, iar coloanele corespunzătoare au elementele situate în afara acestor linii nule; suprimîndu-le, obținem a doua clasă de echivalență.

Vîrful (x_5), rămas, formează a treia clasă de echivalență.

Graful descompus în subgrafe tare conexe este cel din figura II.5.6.

Pe acest graf se pot citi următoarele drumuri hamiltoniene:

$$d_{H_1} = \{ x_1, x_3, x_2, x_4, x_5 \} ,$$
$$d_{H_2} = \{ x_1, x_2, x_4, x_3, x_5 \} .$$

Scanned with OKEN Scanner

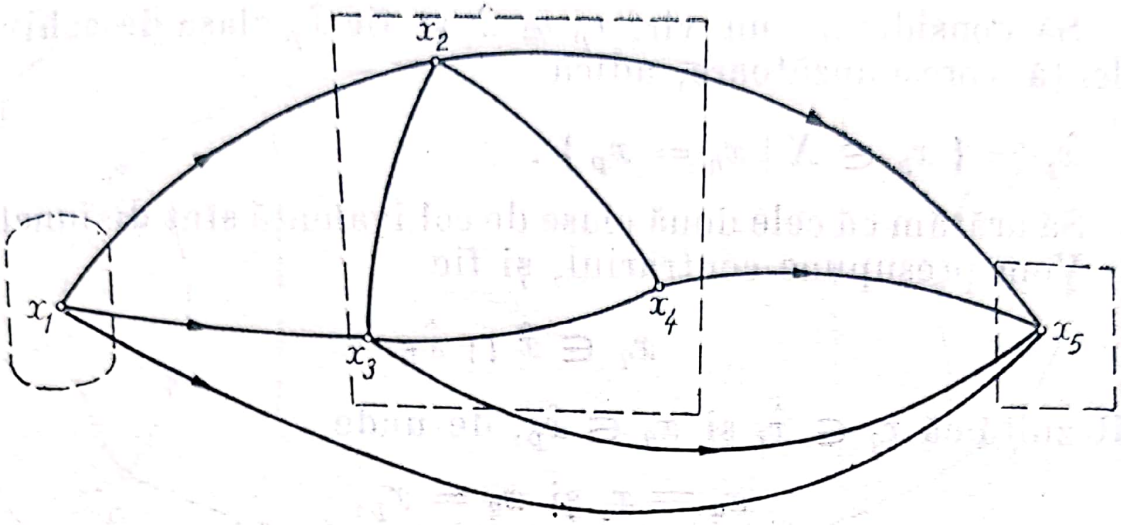


Fig. II.5.6

1°. **Justificarea algoritmului lui Foulkes.** Fie $G = (X, F)$ un graf finit, în care definim următoarea relație de echivalență a vîrfurilor: „vom spune că două vîrfuri x_i și x_k sînt echivalente dacă există un drum de la x_i la x_k și înapoi". Vom nota $x_i = x_k$.

Într-adevăr aceasta este o relație de echivalență, deoarece:

1) $x_i = x_i$, $x_i \in X$

2) din $x_i \equiv x_k$ rezultă și $x_k = x_i$

3) din $x_i \equiv x_k$ și $x_k \equiv x_p$ rezultă $x_i \equiv x_p$,

după cum rezultă și din figura II.5.7.

Relația de echivalență astfel definită (ca orice altă relație de echivalență de altfel) împarte mulțimea vîrfurilor grafului în clase disjuncte de vîrfuri numite *clase de echivalență*.

Vom nota cu $\hat{x}_i$ clasa formată cu vîrfurile grafului echivalente cu vîrful x_i, adică

$$\hat{x}_i = \{ x_j \in X \mid x_j \equiv x_i \} .$$

Fig. II.5.7

Scanned with OKEN Scanner

Să considerăm un vîrf $x_p \notin \hat{x}_i$ și fie $\hat{x}_p$ clasa de echivalență corespunzătoare, adică

$$\hat{x}_p = \{ x_h \in X \mid x_h = x_p \} .$$

Să arătăm că cele două clase de echivalență sînt disjuncte. Vom presupune contrariul, și fie

$$x_q \in \hat{x} \cap \hat{x}_p.$$

Rezultă că $x_q \in \hat{x}_i$ și $x_q \in \hat{x}_p$, de unde

$$x_q \equiv x_i \text{ și } x_q = x_p,$$

iar din 3) rezultă că

$$x_i \equiv x_p \text{ și } x_p \in \hat{x}_i,$$

ceea ce contrazice ipoteza făcută, $x_p \in \hat{x}_i$.

Din definiția grafului tare conex și din modul de constituire a claselor de echivalență rezultă că fiecare clasă de echivalență constituie un graf tare conex.

Să arătăm că vîrfurile a două clase de echivalență distincte nu pot fi legate decît prin drumuri orientate într-un sens și numai unul. Pentru aceasta să presupunem contrariul.

Fie $\hat{x}_i$ și $\hat{x}_j$ două clase disjuncte de vîrfuri. Să presupunem că între vîrfurile $\hat{x}_p \in \hat{x}_i$ și $\hat{x}_q \in \hat{x}_j$ există un drum orientat către $\hat{x}_q$ și să presupunem că există și un drum de la $x_r \in \hat{x}_j$ la $x_s \in \hat{x}_i$ orientat către x_s .

Ținînd seama că fiecare clasă de echivalență constituie un graf tare conex, rezultă că între vîrfurile x_s și x_p există drumuri care le leagă și într-un sens și în altul și, de asemenea, vîrfurile x_r și x_q sînt legate prin drumuri orientate și într-un sens și în altul.

Din considerentele de mai sus rezultă că între vîrfurile $x_p \in \hat{x}_i$ și $\hat{x}_q \in x_j$ există un drum de la x_p către x_q și un drum de la x_q către x_p prin x_r și x_s. Cu alte cuvinte, rezultă că vîrfurile x_p și x_q sînt echivalente și deci x_p și x_q aparțin aceleiași clase de echivalență, fapt ce contrazice ipoteza făcută.

Fie acum graful $G = (X,F)$ și $B = \| b_{ij} \|_n$ matricea booleană asociată grafului.

Scanned with OKEN Scanner

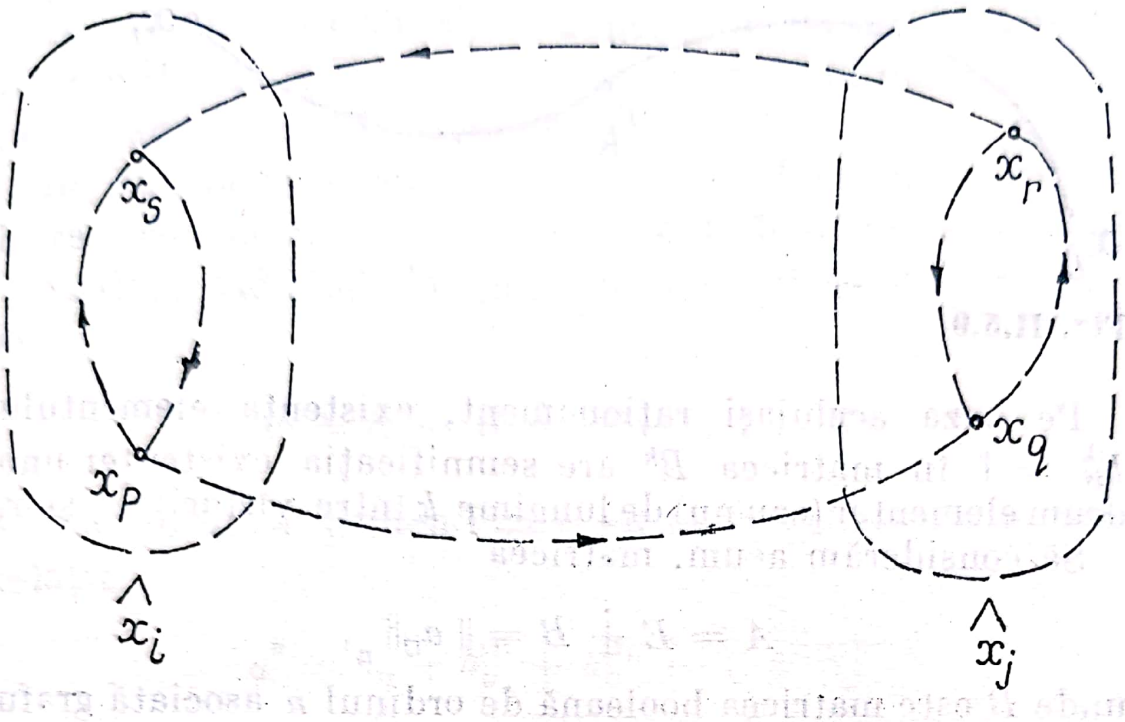


Fig. II.5.8

a) Existența unui element $b_{ij} = 1$ în matricea B indică existența unui arc între vîrfurile x_i și x_j ale grafului dat.

Să considerăm produsul boolean $B \times B = B^2$ și să notăm cu $b_{ij}^{(2)}$ elementul situat pe linia i coloana j din matricea B^2.

Ținînd seama de regulile de înmulțire a două matrice booleene, putem avea

$$b_{ij}^{(2)} = \sum_{k=1}^{n} b_{ik} \cdot b_{kj} = 1,$$

dacă cel puțin unul din produsele $b_{ik} \cdot b_{kj} = 1$.

Pe de altă parte însă,

$$b_{ik} \cdot b_{kj} = 1$$

implică $b_{ik} = 1$ și $b_{kj} = 1$.

Din a) rezultă că între x_i și x_k există un arc și între x_k și x_j există de asemenea un arc.

Prin urmare, existența unui element $b_{ij}^{(2)} = 1$ în matricea B^2 are semnificația existenței în graf a unui drum elementar de lungime 2 între vîrfurile x_i și x_j.

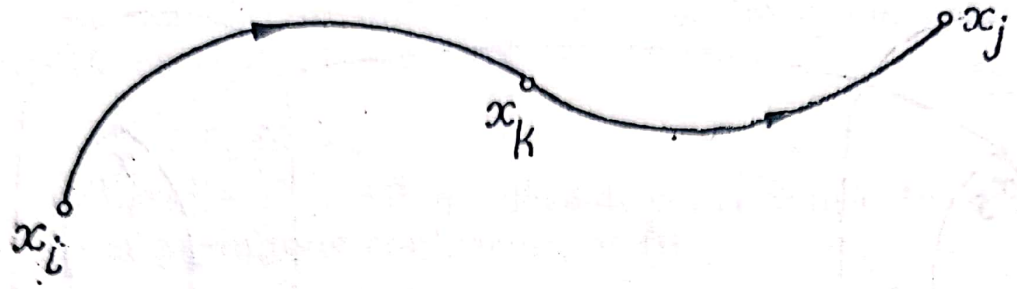


Fig. II.5.9

Pe baza aceluiaşi raţionament, existenţa elementului $b_{ij}^{(k)} = 1$ în matricea B^k are semnificaţia existenţei unui drum elementar (sau nu) de lungime k între vîrfurile x_i şi x_j.

Să considerăm acum, matricea

$$A = E \dot{+} B = \| a_{ij} \|_n,$$

unde B este matricea booleană de ordinul n asociată grafului $G = (X, F)$, cu $X = n$, iar E matricea unitate de acelaşi ordin.

Ridicînd la pătrat boolean matricea A avem

$$A^2 = E \dot{+} B^2 \dot{+} B^2. \qquad (4)$$

Să notăm elementele matricei A^2 cu $a_{ij}^{(2)}$. Presupunem $i \neq j$ şi $a_{ij}^{(2)} = 1$. Din (4) rezultă

$$a_{ij}^{(2)} = e_{ij} \dot{+} b_{ij} \dot{+} b_{ij}^{(2)} = b_{ij} \dot{+} b_{ij}^{(2)},$$

deoarece, pentru $i \neq j$, $e_{ij} = 0$.

Avem $a_{ij}^{(2)} = 1$ dacă şi numai dacă are loc una din următoarele situaţii:

I) $b_{ij} = 1$ şi $b_{ij}^{(2)} = 0$.

În acest caz $a_{ij}^{(2)} = 1$ are semnificaţia existenţei unui drum de lungime 1 între vîrfurile x_i şi x_j.

II) $b_{ij} = 0$ şi $b_{ij}^{(2)} = 1$,

ceea ce înseamnă că $a_{ij}^{(2)} = 1$ semnalează existenţa unui drum de lungime 2 între vîrfurile x_i şi x_j.

III) $b_{ij} = 1$ şi $b_{ij}^{(2)} = 1$.

Scanned with OKEN Scanner

În această situaţie $a_{ij}^{(2)} = 1$ pune în evidenţă existenţa între vîrfurile x_i şi x_j a două drumuri, unul de lungime 1 (adică arc) şi unul de lungime 2.

În concluzie, elementul $a_{ij}^{(2)} = 1$ evidenţiază existenţa între vîrfurile x_i şi x_j a unui drum de lungime cel mult 2.

Analog, ridicînd la puterea k matricea $A = E \dot{+} B$, avem

$$A^k = E \dot{+} B \dot{+} B^2 \dot{+} \ldots \dot{+} B^k.$$

Din faptul că, pentru $i \neq j$, elementul $a_{ij}^{(k)}$ se obţine prin relaţia

$$a_{ij}^k = b_{ij} \dot{+} b_{ij}^{(2)} \dot{+} b_{ij}^{(3)} \dot{+} \ldots \dot{+} b_{ij}^{(k)},$$

rezultă că $a_{ij}^{(k)} = 1$ dacă există un drum (elementar sau nu) de lungime cel mult k între vîrfurile x_i şi x_j ale grafului.

Dacă ţinem seama de faptul că orice drum (elementar sau nu) constă din drumuri parţiale elementare şi că lungimea unui drum elementar într-un graf cu n vîrfuri este cel mult egală cu $n - 1$, rezultă că este suficient ca matricea A să fie ridicată la puterea cel mult $n - 1$.

Pentru micşorarea volumului de calcule se determină puterile:

$$A^2,\ A^4,\ A^8,\ \ldots\ A^{2^k}\ ; \qquad 2^k < n.$$

Dacă

$$A^{2^k} = A^{2^{k+1}}$$

şi dacă vîrfurilor $x_{i_1}, x_{i_2}, \ldots, x_{i_n}$ le corespund, în matricea A^{2^k}, linii formate din elemente egale cu unitatea şi coloane formate numai din 0 (cu excepţia elementului situat la intersecţii), aceste vîrfuri formează prima clasă de echivalenţă.

Scanned with OKEN Scanner

Într-adevăr, din prezența elementelor egale cu unitatea, situate la intersecția liniilor și coloanelor corespunzătoare acestor vîrfuri, rezultă că, oricare ar fi vîrfurile x_{i_p} și x_{i_k}, acestea sînt legate prin drumuri de lungime cel mult $2^k < n - 1$ într-un sens și în altul.

Deoarece elementele situate pe coloanele corespunzătoare acestor vîrfuri sînt toate nule, rezultă că nu există drumuri de la celelalte vîrfuri ale grafului, orientate către vîrfurile $x_{i_1}, \ldots, x_{i_r}$.

Tot din faptul că pe liniile corespunzătoare acestor vîrfuri toate elementele sînt egale cu unitatea rezultă că între aceste vîrfuri și celelalte vîrfuri ale grafului există drumuri de lungime cel mult 2^k, care au vîrful final în afara acestei clase.

Eliminînd aceste vîrfuri din graf, ceea ce revine la a elimina din matricea A^{2^k} liniile și coloanele corespunzătoare, obținem matricea A_1. Acesteia îi aplicăm același procedeu pentru determinarea celei de-a doua clase de echivalență etc.

După stabilirea claselor de echivalență, citirea drumurilor hamiltoniene în graful dat se face stabilind toate drumurile hamiltoniene existente în fiecare clasă și apoi racordîndu-le cu ajutorul drumurilor dintre clase (dacă este posibil).

2°. **Găsirea drumurilor hamiltoniene într-un graf cu ajutorul produsului de matrice latine.** Mai întîi arătăm cum se stabilesc elementele unei matrice latine asociată unui graf.

Fiind dat graful $G = (X,F)$ (figura II.5.10), elementele C_{ij} ale matricei latine asociate acestuia sînt:

$$C_{ij} = \begin{cases} 0, \text{ dacă nu există arc de la } x_i \text{ la } x_j \text{ și dacă } i = j. \\ x_i\, x_j, \text{ dacă există arcul } (x_i, x_j). \end{cases}$$

Scanned with OKEN Scanner

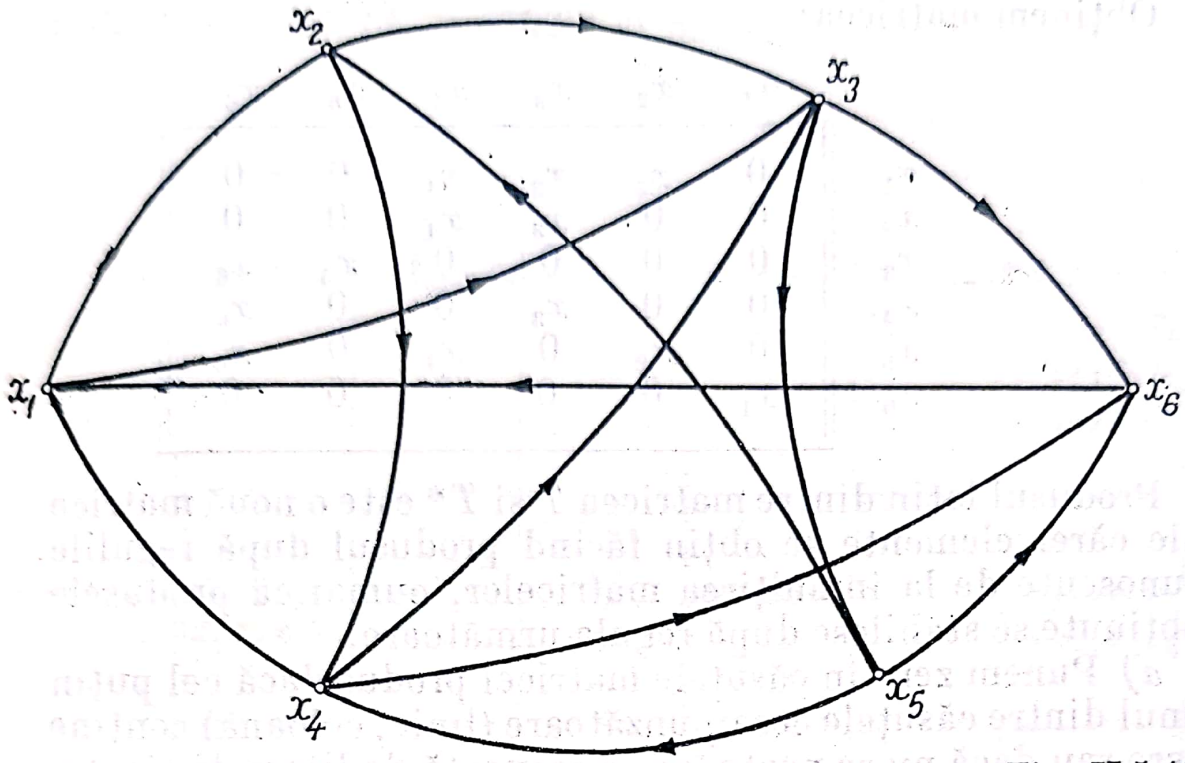


Fig. II.5.10

În exemplul dat avem matricea:

$T =$

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	$x_1,\ x_2$	$x_1,\ x_3$	$x_1,\ x_4$	0	0
x_2	0	0	$x_2,\ x_3$	$x_2,\ x_4$	0	0
x_3	0	0	0	0	$x_3,\ x_5$	$x_3,\ x_6$
x_4	0	0	$x_4,\ x_3$	0	0	$x_4,\ x_6$
x_5	0	$x_5,\ x_2$	0	$x_5,\ x_4$	0	$x_5,\ x_6$
x_6	$x_6,\ x_1$	0	0	0	0	0

Suprimăm în această matrice vîrful inițial din fiecare căsuță aparținînd arcului înscris în ea.

Scanned with OKEN Scanner

Obținem matricea:

$T^* =$

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	x_2	x_3	x_4	0	0
x_2	0	0	x_3	x_4	0	0
x_3	0	0	0	0	x_5	x_6
x_4	0	0	x_3	0	0	x_6
x_5	0	x_2	0	x_4	0	x_6
x_6	x_1	0	0	0	0	0

Produsul latin dintre matricea T și T^* este o nouă matrice ale cărei elemente se obțin făcînd produsul după regulile cunoscute de la înmulțirea matricelor, numai că produsele obținute se stabilesc după regula următoare:

a) Punem zero în căsuțele matricei produs dacă cel puțin unul dintre căsuțele corespunzătoare (linie, coloană) conține zero sau dacă nu se poate face o secvență de litere distincte.

b) În restul căsuțelor punem toate secvențele distincte care apar cu ocazia aceleiași operații.

Aplicînd această regulă și notînd cu L semnul operației de înmulțire latină, produsul latin notat TLT^* va fi:

$T^2 = TLT^* =$

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	0	x_1,x_2,x_3 x_1,x_4,x_3	x_1,x_2,x_4	x_1,x_3,x_5	x_1,x_3,x_6 x_1,x_4,x_6
x_2	0	0	x_2,x_4,x_3	0	x_2,x_3,x_5	x_2,x_3,x_6 x_2,x_4,x_6
x_3	x_3,x_6,x_1	x_3,x_5,x_2	0	x_3,x_5,x_4	0	x_3,x_5,x_6
x_4	x_4,x_6,x_1				x_4,x_3,x_5	x_4,x_3,x_6
x_5	x_5,x_6,x_1	0	x_5,x_4,x_3 x_5,x_2,x_3	x_5,x_2,x_4		x_5,x_4,x_6
x_6	0	x_6,x_1,x_2	x_6,x_1,x_3	x_6,x_1,x_4	0	0

Scanned with OKEN Scanner

Calculăm apoi $T^3 = T^2LT^*$ şi obţinem:

$T^3 =$

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	$x_1x_3x_5x_2$	$x_1x_2x_4x_5$	$x_1x_3x_5x_4$	$x_1x_2x_3x_5$ $x_1x_4x_3x_5$	$x_1x_2x_3x_6$ $x_1x_3x_5x_6$ $x_1x_4x_3x_6$ $x_5x_2x_4x_6$
x_2	$x_2x_3x_6x_1$ $x_2x_4x_6x_1$	0	0	$x_2x_3x_5x_4$	$x_2x_4x_3x_5$	$x_2x_4x_3x_6$ $x_2x_3x_5x_6$
x_3	$x_3x_5x_6x_1$	$x_3x_6x_1x_2$	0	$x_3x_6x_1x_4$ $x_3x_5x_2x_4$	0	$x_3x_5x_4x_6$
x_4	$x_4x_3x_5x_2$	$x_4x_6x_1x_3$ $x_4x_6x_1x_2$	$x_4x_6x_1x_3$	0	0	$x_4x_3x_5x_6$
x_5	$x_5x_4x_6x_1$	$x_5x_6x_1x_2$	$x_5x_6x_1x_3$ $x_5x_2x_4x_3$	$x_5x_6x_1x_4$	0	$x_5x_4x_3x_6$ $x_5x_2x_3x_6$ $x_5x_2x_4x_6$
x_6	0	0	$x_6x_1x_2x_3$ $x_6x_1x_4x_3$	$x_6x_1x_2x_4$	$x_6x_1x_3x_5$	0

În continuare vom avea: $T^4 = T^3\,LT^*$:

Scanned with OKEN Scanner

$T^4 =$

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	$x_1x_4x_3x_5x_2$	0	$x_1x_3x_5x_2x_4$ $x_1x_2x_3x_5x_4$	$x_1x_2x_4x_3x_5$	$x_1x_4x_3x_5x_6$ $x_1x_2x_4x_3x_6$ $x_1x_2x_3x_5x_6$ $x_1x_3x_5x_4x_6$
x_2	$x_2x_4x_3x_6x_1$ $x_2x_3x_5x_6x_1$	0	$x_2x_4x_6x_1x_3$	$x_2x_3x_6x_1x_4$	0	$x_2x_3x_5x_4x_6$ $x_2x_4x_3x_5x_6$
x_3	$x_3x_5x_4x_6x_1$	$x_3x_5x_6x_1x_2$	0	$x_3x_5x_6x_1x_4$ $x_3x_6x_1x_2x_4$	0	$x_3x_5x_2x_4x_6$
x_4	$x_4x_3x_5x_6x_1$	$x_4x_3x_6x_1x_2$	$x_4x_6x_1x_2x_3$	0	$x_4x_6x_1x_3x_5$	0
x_5	$x_5x_2x_4x_6x_1$ $x_5x_4x_3x_6x_1$ $x_5x_2x_3x_6x_1$	$x_5x_4x_6x_1x_2$	$x_5x_4x_6x_1x_3$ $x_5x_6x_1x_2x_4$ $x_5x_6x_1x_4x_3$	$x_5x_6x_1x_2x_4$	0	$x_5x_2x_4x_3x_6$
x_6	0	$x_6x_1x_3x_5x_2$	$x_6x_1x_2x_4x_3$	$x_6x_1x_3x_5x_4$	$x_6x_1x_2x_3x_5$ $x_6x_1x_4x_3x_5$	0

Scanned with OKEN Scanner

$T^5 =$

x_1	x_2	x_3	x_4	x_5	x_6
0	0	0	0	0	$x_1x_2x_3x_5x_4x_6$ $x_1x_2x_4x_3x_5x_6$ $x_1x_3x_5x_2x_4x_6$
$x_2x_3x_5x_4x_6x_1$ $x_2x_4x_3x_5x_6x_1$	0	0	$x_2x_3x_5x_6x_1x_4$	$x_2x_4x_6x_1x_3x_5$	0
$x_3x_5x_2x_4x_6x_1$	$x_3x_5x_4x_6x_1x_2$	0	$x_3x_5x_6x_1x_2x_4$	0	0
0	$x_4x_3x_5x_6x_1x_2$ $x_4x_6x_1x_3x_5x_2$	0	0	$x_4x_6x_1x_2x_3x_5$	0
$x_5x_2x_4x_3x_6x_1$	$x_5x_4x_3x_6x_1x_2$	$x_5x_2x_4x_6x_1x_3$ $x_5x_4x_6x_1x_2x_3$ $x_5x_6x_1x_2x_4x_3$	$x_5x_2x_3x_6x_1x_4$	0	0
0	$x_6x_1x_4x_3x_5x_2$	0	$x_6x_1x_3x_5x_2x_4$ $x_6x_1x_2x_3x_5x_4$	$x_6x_1x_2x_4x_3x_5$	0

Scanned with OKEN Scanner

În matricea $T^5 = T^4LT^*$ găsim drumurile hamiltoniene ale grafului dat și anume:

$$
\begin{aligned}
d_H^1 &= (x_1, x_2, x_3, x_5, x_4, x_6),\\
d_H^2 &= (x_1, x_2, x_4, x_3, x_5, x_6),\\
d_H^3 &= (x_1, x_3, x_5, x_2, x_4, x_6),\\
d_H^4 &= (x_2, x_3, x_5, x_4, x_6, x_1),\\
d_H^5 &= (x_2, x_4, x_3, x_5, x_6, x_1)\\
d_H^6 &= (x_2, x_3, x_5, x_6, x_1, x_4)\\
d_H^7 &= (x_2, x_4, x_6, x_1, x_3, x_5)\\
d_H^8 &= (x_3, x_5, x_2, x_4, x_6, x_1)\\
d_H^9 &= (x_3, x_5, x_4, x_6, x_1, x_2),\\
d_H^{10} &= (x_3, x_5, x_6, x_1, x_2, x_4),\\
d_H^{11} &= (x_4, x_3, x_5, x_6, x_1, x_2),\\
d_H^{12} &= (x_4, x_6, x_1, x_3, x_5, x_2),\\
d_H^{13} &= (x_4, x_6, x_1, x_2, x_3, x_5),\\
d_H^{14} &= (x_5, x_2, x_4, x_3, x_6, x_1),\\
d_H^{15} &= (x_5, x_4, x_3, x_6, x_1, x_2),\\
d_H^{16} &= (x_5, x_2, x_4, x_6, x_1, x_3),\\
d_H^{17} &= (x_5, x_4, x_6, x_1, x_2, x_3),\\
d_H^{18} &= (x_5, x_6, x_1, x_2, x_4, x_3),\\
d_H^{19} &= (x_6, x_1, x_4, x_3, x_5, x_2),\\
d_H^{20} &= (x_6, x_1, x_3, x_5, x_2, x_4),\\
d_H^{21} &= (x_6, x_1, x_2, x_3, x_5, x_4),\\
d_H^{22} &= (x_6, x_1, x_2, x_4, x_3, x_5),\\
d_H^{23} &= (x_5, x_2, x_3, x_6, x_1, x_4).
\end{aligned}
$$

Observații: Din cele expuse mai înainte ajungem la concluzia că nu este necesar să calculăm puterile succe ive de forma T^kLT, ci putem obține, de exemplu, matricea T^{2p}, efectuînd produsul

$$T^{2p} = T^pL\,(T^p)^*,$$

unde $(T^p)^*$ se obține din T^p prin suprimarea primului vîrf din secvențele ce le conține.

Scanned with OKEN Scanner

§ 6. APLICAȚIE

Să considerăm o întreprindere industrială care printre alte secții are și o secție de cazangerie.

Secția cazangerie constituie secția de bază a întreprinderii, deoarece are ponderea cea mai mare în volumul producției. În această secție se fabrică următoarele produse principale comparabile: boilere, hidrofoare, rezervoare, aparate contra curent. Acestea, la rîndul lor, sînt de diferite tipuri, totalizînd 14 produse diferite

P_1 — hidrofoare cu capacitate de 300 l,
P_2 — hidrofoare cu capacitate de 1500 l,
P_3 — hidrofoare cu capacitate de 3000 l,
P_4 — aparate contra curent cu 8 tronsoane și cu țevi de 3 m,
P_5 — aparate contra curent cu 6 tronsoane și cu țevi 3 m,
P_6 — aparate contra curent cu 4 m tronsoane și cu țevi de 25 m,
P_7 — rezervoare cu o capacitate de 500 l,
P_8 — rezervoare cu o capacitate de 2000 l,
P_9 — rezervoare cu o capacitate de 1000 l,
P_{10} — rezervoare cu o capacitate de 5000 l,
P_{11} — boilere de 200 l,
P_{12} — boilere de 600 l,
P_{13} — boilere de 2000 l,
P_{14} — boilere de 5000 l.

În afară de aceste produse principale comparabile, în întreprindere se fabrică și alte produse secundare necomparabile. Producția comparabilă deține 80% din întregul volum al producției. Presupunem că din evidențele întreprinderii se cunosc costurile de trecere de la fabricarea unui produs la celelalte. Ne propunem să determinăm ordinea optimă

Scanned with OKEN Scanner

de lansare în fabricaţie a celor 14 tipuri de produse, astfel încît costul total al trecerilor să fie minim.

O asemenea problemă se reduce la determinarea drumurilor hamiltoniene în graful în care vîrfurile reprezintă cele 14 produse, arcele reprezintă posibilităţile de trecere de la un produs la altul în procesul de fabricaţie, iar lungimile arcelor reprezintă cheltuielile implicate de acestetreceri.

Din mulţimea drumurilor hamiltoniene existente în graful astfel construit, alegem pe acela care are lungimea totală minimă, ceea ce corespunde unei ordini de lansare în producţie, în care costul total este minim.

Pentru rezolvarea acestei probleme folosim algoritmul lui Foulkes.

Pentru aplicarea algoritmului lui Foulkes în rezolvarea acestei probleme procedăm după cum urmează:

a) asociem grafului o matrice M cu elemente

$$r_{ij} = \begin{cases} C_{ij} = (x_i, x_j), \text{ dacă există arcul } (x_i, x_j), \\ \infty, \text{ dacă nu există arcul } (x_i, x_j), \\ 0, \text{ dacă } i = j. \end{cases}$$

Matricei M îi asociem o matrice booleană B, ale cărei elemente b_{ij} se stabilesc după regula:

$$b_{ij} = \begin{cases} 1, \text{ dacă } C_{ij} < C_{ji}, \\ 0, \text{ dacă } C_{ij} > C_{ji} \text{ sau dacă } i = j. \end{cases}$$

Cu alte cuvinte, pentru găsirea elementelor matricei booleene comparăm elementele simetrice faţă de diagonala principală, înlocuindu-se elementul mai mic cu unitatea şi elementul mai mare cu zero.

Această alegere este justificată de faptul că se urmăreşte găsirea drumului hamiltonian de lungime minimă.

Există situaţii în care două sau mai multe elemente simetrice faţă de diagonală să fie egale. În acest caz,unitatea poate fi pusă în locul oricăreia dintre ele, urmînd ca cea rămasă să fie înlocuită cu zero,

Scanned with OKEN Scanner

$M =$

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}
P_1	0	50	83	1018	635	287	13	25	19	75	19	29	94	189
P_2	15	0	80	1014	631	285	11	24	17	71	17	32	91	195
P_3	14	43	0	1016	626	272	10	21	20	73	20	27	98	190
P_4	16	48	83	0	615	270	12	23	21	79	21	30	95	185
P_5	19	50	85	997	0	275	15	27	23	75	23	25	97	187
P_6	17	15	82	1002	625	0	9	25	16	77	16	29	92	188
P_7	10	41	78	998	632	281	0	22	18	71	18	31	99	194
P_8	18	42	76	1005	617	283	14	0	19	76	79	26	91	191
P_9	19	44	79	1010	630	285	13	23	0	74	22	24	95	184
P_{10}	11	46	81	1011	638	276	10	26	22	0	16	27	93	183
P_{11}	13	40	75	1015	635	289	11	24	16	75	0	29	96	187
P_{12}	12	46	81	1016	618	295	13	19	19	72	19	0	94	190
P_{13}	13	47	80	1012	620	277	12	21	20	78	20	31	0	193
P_{14}	17	50	83	1013	632	283	14	20	16	73	16	28	91	0

Matricea booleană este:

$B =$

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}
P_1	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P_2	1	0	0	0	0	0	1	1	1	0	1	1	0	0
P_3	1	1	0	0	0	0	1	1	1	1	1	1	0	0
P_4	1	1	1	0	0	1	1	1	1	1	1	1	1	1
P_5	1	1	1	0	1	1	1	1	1	1	1	1	1	1
P_6	1	1	1	0	0	0	1	1	1	1	1	1	1	1
P_7	1	0	0	0	0	0	0	0	0	0	0	0	0	0
P_8	1	0	0	0	0	0	1	0	1	0	1	0	0	0
P_9	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P_{10}	1	1	0	0	0	0	1	1	1	0	0	1	0	0
P_{11}	1	0	0	0	0	0	1	0	1	0	1	0	0	0
P_{12}	1	0	0	0	0	0	1	1	1	0	1	0	0	0
P_{13}	1	1	1	0	0	0	1	1	1	1	1	1	0	0
P_{14}	1	1	1	0	0	0	1	1	1	1	1	1	1	0

Scanned with OKEN Scanner

Pentru aplicarea algoritmului lui Foulkes calculăm matricea $A = B + E$:

$A = B + E =$

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}
P_1	1	0	0	0	0	0	0	0	1	0	0	0	0	0
P_2	1	1	0	0	0	0	1	1	1	0	1	1	0	0
P_3	1	1	1	0	0	0	1	1	1	1	1	1	0	0
P_4	1	1	1	1	1	1	1	1	1	1	1	1	1	1
P_5	1	1	1	0	1	1	1	1	1	1	1	1	1	1
P_6	1	1	1	0	0	1	1	1	1	1	1	1	1	1
P_7	1	0	0	0	0	0	1	0	0	0	0	0	0	0
P_8	1	0	0	0	0	0	1	1	1	0	1	0	0	0
P_9	0	0	0	0	0	0	1	0	1	0	0	0	0	0
P_{10}	1	1	0	0	0	0	1	1	1	1	1	1	0	0
P_{11}	1	0	0	0	0	0	1	0	1	0	1	0	0	0
P_{12}	1	0	0	0	0	0	1	1	1	0	1	1	0	0
P_{13}	1	1	1	0	0	0	1	1	1	1	1	1	1	1

de unde:

$A^2 =$

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}
P_1	1	0	0	0	0	0	1	0	1	0	0	0	0	0
P_2	1	1	0	0	0	0	1	1	1	0	1	1	0	0
P_3	1	1	1	0	0	0	1	1	1	1	1	1	0	0
P_4	1	1	1	1	1	1	1	1	1	1	1	1	1	1
P_5	1	1	1	1	1	1	1	1	1	1	1	1	1	1
P_6	1	1	1	0	0	1	1	1	1	1	1	1	1	1
P_7	1	0	0	0	0	0	1	0	1	0	0	0	0	0
P_8	1	0	0	0	0	0	1	1	1	0	1	0	0	0
P_9	1	0	0	0	0	0	1	0	1	0	0	0	0	0
P_{10}	1	1	0	0	0	0	1	1	1	1	1	1	0	0
P_{11}	1	0	0	0	0	0	1	0	1	0	1	0	0	0
P_{12}	1	0	0	0	0	0	1	1	1	0	1	1	0	0
P_{13}	1	1	1	0	0	0	1	1	1	1	1	1	1	0
P_{14}	1	1	1	0	0	0	1	1	1	1	1	1	1	1

Scanned with OKEN Scanner

$A^3 =$

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}
P_1	1	0	0	0	0	0	1	0	1	0	0	0	0	0
P_2	1	1	0	0	0	0	1	1	1	0	1	1	0	0
P_3	1	1	1	0	0	0	1	1	1	1	1	1	0	0
P_4	1	1	1	1	1	1	1	1	1	1	1	1	1	1
P_5	1	1	1	1	1	1	1	1	1	1	1	1	1	1
P_6	1	1	1	0	0	1	1	1	1	1	1	1	1	1
P_7	1	0	0	0	0	0	1	0	1	0	0	0	0	0
P_8	1	0	0	0	0	0	1	1	1	0	1	0	0	0
P_9	1	0	0	0	0	0	1	0	1	0	0	0	0	0
P_{10}	1	0	0	0	0	0	1	1	1	1	1	1	0	0
P_{11}	1	0	0	0	0	0	1	0	1	0	1	0	0	0
P_{12}	1	0	0	0	0	0	1	1	1	0	1	1	0	0
P_{13}	1	1	1	0	0	0	1	1	1	1	1	1	1	0
P_{14}	1	1	1	0	0	0	1	1	1	1	1	1	1	1

Deci clasa I este formată din produsele P_4 și P_5.

Notăm cu A_1 matricea rămasă și procedăm la fel cum am procedat cu matricea A.

$A_1 =$

	P_1	P_2	P_3	P_4	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}
P_1	1	0	0	0	1	0	1	0	0	0	0	0
P_2	1	1	0	0	1	1	1	0	1	1	0	0
P_3	1	1	1	0	1	1	1	1	1	1	0	0
P_4	1	1	1	1	1	1	1	1	1	1	1	1
P_7	1	0	0	0	1	0	1	0	0	0	0	0
P_8	1	0	0	0	1	1	1	0	1	0	0	0
P_9	1	0	0	0	1	0	1	0	0	0	0	0
P_{10}	1	1	0	0	1	1	1	1	1	1	0	0
P_{11}	1	0	0	0	1	0	1	0	1	0	0	0
P_{12}	1	0	0	0	1	1	1	0	1	1	0	0
P_{13}	1	1	1	0	1	1	1	1	1	1	1	0
P_{14}	1	1	1	0	1	1	1	1	1	1	1	1

Scanned with OKEN Scanner

$A_1^2 =$

	P_1	P_2	P_3	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}
P_1	1	0	0	0	1	0	1	0	0	0	0	0
P_2	1	1	0	0	1	1	1	0	1	1	0	0
P_3	1	1	1	0	1	1	1	1	1	1	0	0
P_4	1	1	1	1	1	1	1	1	1	1	1	1
P_6	1	0	0	0	1	0	1	0	0	0	0	0
P_8	1	0	0	0	1	1	1	0	1	0	0	0
P_9	1	0	0	0	1	0	1	0	0	0	0	0
P_{10}	1	1	0	0	1	1	1	1	1	1	0	0
P_{11}	1	0	0	0	1	0	1	0	1	0	0	0
P_{12}	1	0	0	0	1	1	1	0	1	1	0	0
P_{13}	1	1	1	0	1	1	1	1	1	1	1	0
P_{14}	1	1	1	0	1	1	1	1	1	1	1	1

Clasa a II-a, P_6

Notăm cu A_2 matricea rămasă din A_1^2 prin suprimarea liniei P_6 și a coloanei P_6.

$A_2 =$

	P_1	P_2	P_3	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}
P_1	1	0	0	1	0	1	0	0	0	0	0
P_2	1	1	0	1	1	1	0	1	1	0	0
P_3	1	1	1	1	1	1	1	1	1	0	0
P_7	1	0	0	1	0	1	0	0	0	0	0
P_8	1	0	0	1	1	1	0	1	0	0	0
P_9	1	0	0	1	0	1	0	0	0	0	0
P_{10}	1	1	0	1	1	1	1	1	1	0	0
P_{11}	1	0	0	1	0	1	0	1	0	0	0
P_{12}	1	0	0	1	1	1	0	1	1	0	0
P_{13}	1	1	1	1	1	1	1	1	1	1	0
P_{14}	1	1	1	1	1	1	1	1	1	1	1

Scanned with OKEN Scanner

$A_2^2 =$

	P_1	P_2	P_3	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}
P_1	1	0	0	1	0	1	0	0	0	0	0
P_2	1	1	0	1	1	1	0	1	1	0	0
P_3	1	1	1	1	1	1	1	1	1	0	0
P_7	1	0	0	1	0	1	0	0	0	0	0
P_8	1	0	0	1	1	1	0	1	0	0	0
P_9	1	0	0	1	0	1	0	0	0	0	0
P_{10}	1	1	0	1	1	1	1	1	1	0	0
P_{11}	1	0	0	1	0	1	0	1	0	0	0
P_{12}	1	0	0	1	1	1	0	1	1	0	0
P_{13}	1	1	1	1	1	1	1	1	1	1	0
P_{14}	1	1	1	1	1	1	1	1	1	1	1

Clasa a III-a, P_{14}

Notăm cu A_3 matricea rămasă din A_2^2 prin suprimarea liniei P_{14} și a coloanei P_{14}

$A_3 =$

	P_1	P_2	P_3	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}
P_1	1	0	0	1	0	1	0	0	0	0
P_2	1	1	0	1	1	1	0	1	1	0
P_3	1	1	1	1	1	1	1	1	1	0
P_7	1	0	0	1	0	1	0	0	0	0
P_8	1	0	0	1	1	1	0	1	0	0
P_9	1	0	0	1	0	1	0	0	0	0
P_{10}	1	1	0	1	1	1	1	1	1	0
P_{11}	1	0	0	1	0	1	0	1	0	0
P_{12}	1	0	0	1	1	1	0	1	1	0
P_{13}	1	1	1	1	1	1	1	1	1	1

Scanned with OKEN Scanner

$A_3^2 =$

	P_1	P_2	P_3	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}
P_1	1	0	0	1	0	1	0	0	0	0
P_2	1	1	0	1	1	1	0	1	1	0
P_3	1	1	1	1	1	1	1	1	1	0
P_7	1	0	0	1	0	1	0	0	0	0
P_8	1	0	0	1	1	1	0	1	0	0
P_9	1	0	0	1	0	1	0	0	0	0
P_{10}	1	1	0	1	1	1	1	1	1	0
P_{11}	1	1	0	1	1	1	1	1	1	0
P_{12}	1	0	0	1	1	1	0	1	1	0
P_{13}	1	1	1	1	1	1	1	1	1	1

Clasa a IV-a, P_{13}

Notăm cu A_4 matricea rămasă din matricea A_3^2, prin suprimarea liniei P_{13} și a coloanei P_{13}.

$A_4 =$

	P_1	P_2	P_3	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
P_1	1	0	0	1	0	1	0	0	0
P_2	1	1	0	1	1	1	0	1	1
P_3	1	1	1	1	1	1	1	1	1
P_7	1	0	0	1	0	1	0	0	0
P_8	1	0	0	1	1	1	0	1	0
P_9	1	0	0	1	0	1	0	0	0
P_{10}	1	1	0	1	1	1	1	1	1
P_{11}	1	0	0	1	0	1	0	1	0
P_{12}	1	0	0	1	1	1	0	1	1

$A_4^2 =$

	P_1	P_2	P_3	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
P_1	1	0	0	1	0	1	0	0	0
P_2	1	1	0	1	1	1	0	1	1
P_3	1	1	1	1	1	1	1	1	1
P_7	1	0	0	1	0	1	0	0	0
P_8	1	0	0	1	1	1	0	1	0
P_9	1	0	0	1	0	1	0	0	0
P_{10}	1	1	0	1	1	1	1	1	1
P_{11}	1	0	0	1	0	1	0	1	0
P_{12}	1	0	0	1	1	1	0	1	1

Scanned with OKEN Scanner

Clasa a V-a, P_3

Notăm cu A_5 matricea rămasă din A_4^2 prin suprimarea liniei P_3 și a coloanei P_3.

$$A_5 = \begin{array}{c|cccccccc} & P_1 & P_2 & P_7 & P_8 & P_9 & P_{10} & P_{11} & P_{12} \\ \hline P_1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ P_2 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ P_7 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ P_8 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ P_9 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ P_{10} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ P_{11} & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ P_{12} & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{array}$$

$$A_5^2 = \begin{array}{c|cccccccc} & P_1 & P_2 & P_7 & P_8 & P_9 & P_{10} & P_{11} & P_{12} \\ \hline P_1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ P_2 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ P_7 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ P_8 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ P_9 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ \cancel{P_{10}} & \cancel{1} & \cancel{1} & \cancel{1} & \cancel{1} & \cancel{1} & \cancel{1} & \cancel{1} & \cancel{1} \\ P_{11} & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ P_{12} & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{array}$$

Clasa a VI-a, P_{10}

La fel obținem:

$$A_6 = \begin{array}{c|ccccccc} & P_1 & P_2 & P_7 & P_8 & P_9 & P_{11} & P_{12} \\ \hline P_1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ P_2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ P_7 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ P_8 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ P_9 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ P_{11} & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ P_{12} & 1 & 0 & 1 & 1 & 1 & 1 & 1 \end{array}$$

Scanned with OKEN Scanner

$$A_6^2 = \begin{array}{c|ccccccc} & P_1 & P_2 & P_7 & P_8 & P_9 & P_{11} & P_{12} \\ P_1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ P_2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ P_7 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ P_8 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ P_9 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ P_{11} & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ P_{12} & 1 & 0 & 1 & 1 & 1 & 1 & 1 \end{array}$$

Clasa a VII-a, P_2

$$A_7 = \begin{array}{c|cccccc} & P_1 & P_7 & P_8 & P_9 & P_{11} & P_{12} \\ P_1 & 1 & 1 & 0 & 1 & 0 & 0 \\ P_7 & 1 & 1 & 0 & 1 & 0 & 0 \\ P_8 & 1 & 1 & 1 & 1 & 1 & 0 \\ P_9 & 1 & 1 & 0 & 1 & 0 & 0 \\ P_{11} & 1 & 1 & 0 & 1 & 1 & 0 \\ P_{12} & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

$$A_7^2 = \begin{array}{c|cccccc} & P_1 & P_7 & P_8 & P_9 & P_{11} & P_{12} \\ P_1 & 1 & 1 & 0 & 1 & 0 & 0 \\ P_7 & 1 & 1 & 0 & 1 & 0 & 0 \\ P_8 & 1 & 1 & 1 & 1 & 1 & 0 \\ P_9 & 1 & 1 & 0 & 1 & 0 & 0 \\ P_{11} & 1 & 1 & 0 & 1 & 1 & 0 \\ P_{12} & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

Clasa a VIII-a, P_{12}

$$A_8 = \begin{array}{c|ccccc} & P_1 & P_7 & P_8 & P_9 & P_{11} \\ P_1 & 1 & 1 & 0 & 1 & 0 \\ P_7 & 1 & 1 & 0 & 1 & 0 \\ P_8 & 1 & 1 & 1 & 1 & 1 \\ P_9 & 1 & 1 & 0 & 1 & 0 \\ P_{11} & 1 & 1 & 0 & 1 & 1 \end{array}$$

Scanned with OKEN Scanner

$$A_8^2 = \begin{array}{c} \\ P_1 \\ P_7 \\ P_8 \\ P_9 \\ P_{11} \end{array} \begin{array}{c} \begin{array}{ccccc} P_1 & P_7 & P_8 & P_9 & P_{11} \end{array} \\ \left\| \begin{array}{ccccc} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{array} \right\| \end{array}$$

Clasa a IX-a, P_8

$$A_9 = \begin{array}{c} \\ P_1 \\ P_7 \\ P_9 \\ P_{11} \end{array} \begin{array}{c} \begin{array}{cccc} P_1 & P_7 & P_9 & P_{11} \end{array} \\ \left\| \begin{array}{cccc} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{array} \right\| \end{array} \quad A_9^2 = \begin{array}{c} \\ P_1 \\ P_7 \\ P^9 \\ P_{11} \end{array} \begin{array}{c} \begin{array}{cccc} P_1 & P_7 & P_9 & P_{11} \end{array} \\ \left\| \begin{array}{cccc} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{array} \right\| \end{array}$$

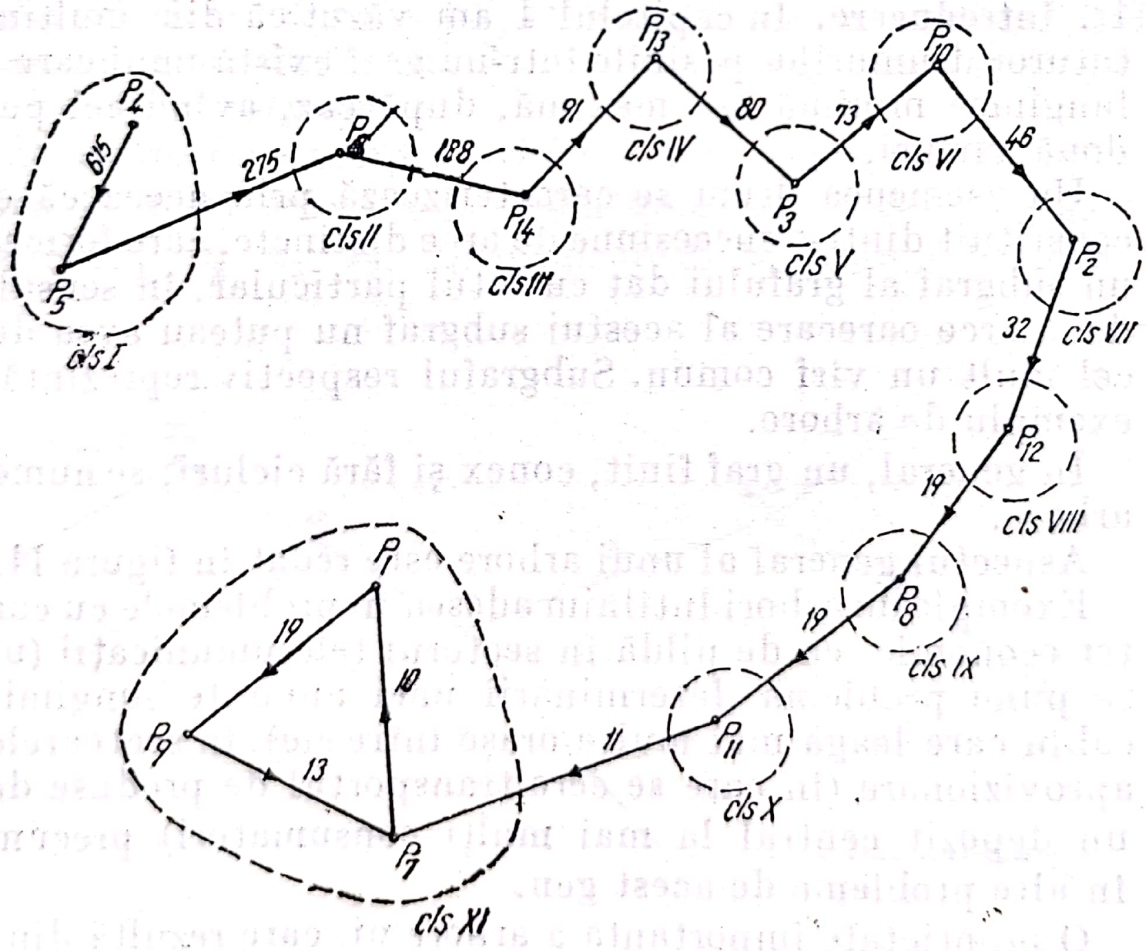


Fig. II.6.1

Clasa a X-a, P_{11}

Ultima clasă, a XI-a, este formată din produsele P_1, P_7, P_9.

Figurînd clasele de echivalenţă rezultate pe baza algoritmului Foulkes, obţinem graful care indică ordinea de lansare în fabricaţie a produselor (figura II.6.1).

Pe acest ultim graf sînt trecute costurile corespunzătoare operaţiilor de trecere.

Costul total de trecere corespunzător drumului hamiltonian de lungime minimă este:

$$\min\ (C_{d_h}) = 615 + 275 + 188 + 91 + 80 + 73 + 46 + 32 + 19 + 19 + 11 + 10 + 11 + 13 = 1491.$$

§ 7. ARBORI ÎNTR-UN GRAF. ARBORE MINIMAL

1°. **Introducere.** În capitolul I am văzut că din mulţimea tuturor drumurilor posibile într-un graf există unul care are lungimea minimă sau maximă, după caz, avînd cel puţin două vîrfuri.

Un asemenea drum se caracterizează prin aceea că este constituit dintr-o succesiune de arce distincte, care formează un subgraf al grafului dat cu totul particular, în sensul că două arce oarecare al acestui subgraf nu puteau avea decît cel mult un vîrf comun. Subgraful respectiv reprezintă un exemplu de arbore.

În general, un graf finit, conex şi fără cicluri, se numeşte arbore.

Aspectul general al unui arbore este redat în figura II.7.1

Exemple de arbori întîlnim adesea în problemele cu caracter economic, ca de pildă în sectorul telecomunicaţii (unde se pune problema determinării unei anumite lungimi de cablu care leagă mai multe oraşe între ele), în sectoarele de aprovizionare (în care se cere transportul de produse de la un depozit central la mai mulţi consumatori) precum şi în alte probleme de acest gen.

O proprietate importantă a arborelui, care rezultă din însăşi definiţia lui, este aceea că suprimînd un arc oarecare al

Scanned with OKEN Scanner

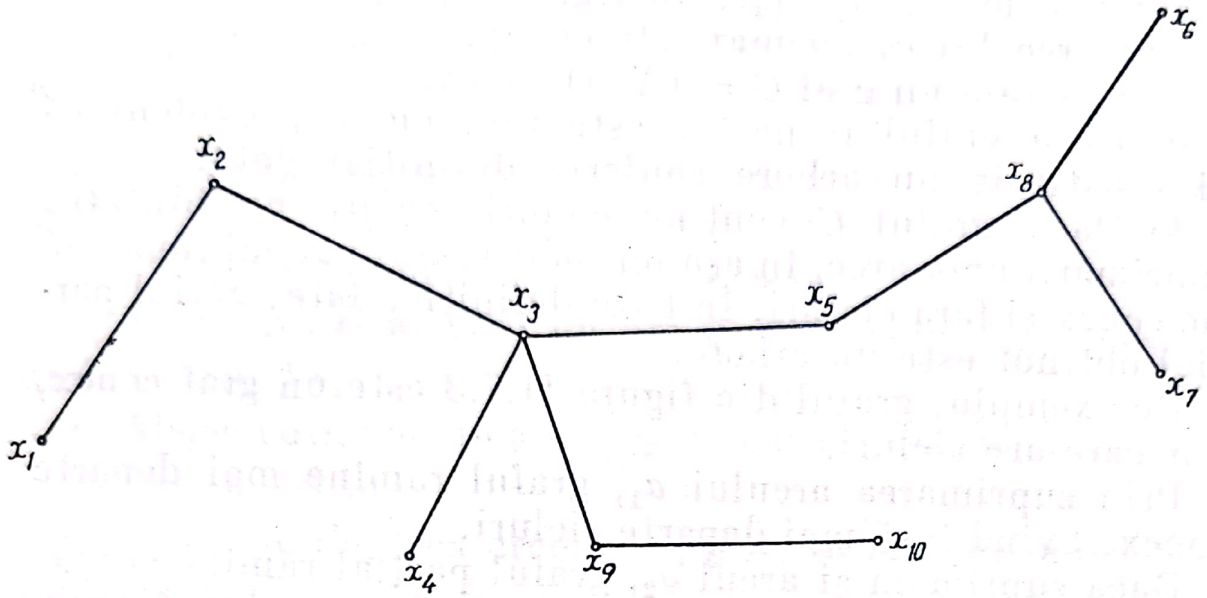


Fig. II.7.1

acestuia, el își pierde proprietatea de a mai fi conex. În legătură cu această proprietate avem teorema:

Un graf $G = (X,F) = (X,A)$ admite un arbore, avînd aceeași mulțime de vîrfuri X dar arce mai puține, dacă și numai dacă el este conex.

Considerăm de exemplu arborele din figura II.7.2

Dacă în acest arbore suprimăm arcul a_4, oricare dintre vîrfurile (x_1, x_2, x_3, x_7, x_8) nu poate fi legat de nici unul

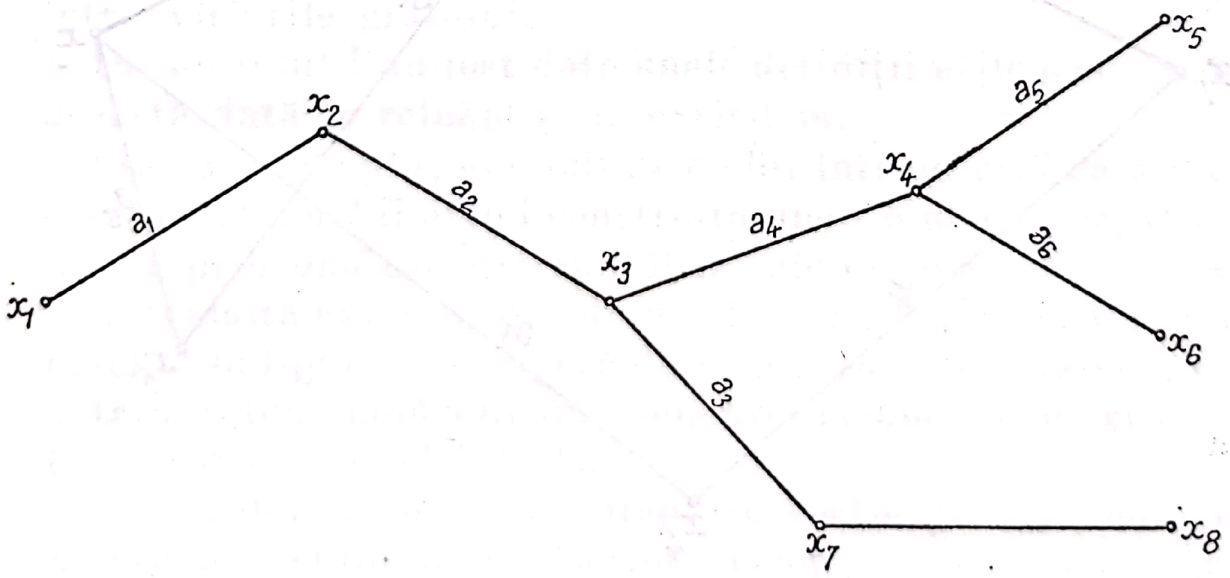


Fig. II.7.2

din vîrfurile (x_4, x_5, x_6); deci graful parțial obținut prin suprimarea lui a_4 nu mai este conex.

Considerăm un graf $G = (X,A)$ conex.

a) Dacă graful respectiv este fără cicluri, evident că el constituie un arbore conform definiției date.

b) Dacă graful G conține cicluri, există posibilitatea suprimării unor arce, în așa fel încît graful parțial rămas să fie conex și fără cicluri. În baza definiției date, graful parțial obținut este un arbore.

De exemplu, graful din figura II.7.3 este un graf conex, dar care are cicluri.

Prin suprimarea arcului a_1, graful rămîne mai departe conex, avînd însă mai departe cicluri.

Dacă suprimăm și arcul a_2, graful parțial rămîne conex, dar de această dată el nu mai are cicluri. Atunci graful parțial rămas este un arbore în virtutea definiției date arborelui.

Observații: Evident, dacă graful $G = (X,A)$ nu este conex, cu atît mai mult grafele parțiale ale acestuia nu sînt conexe și deci G nu conține arbori.

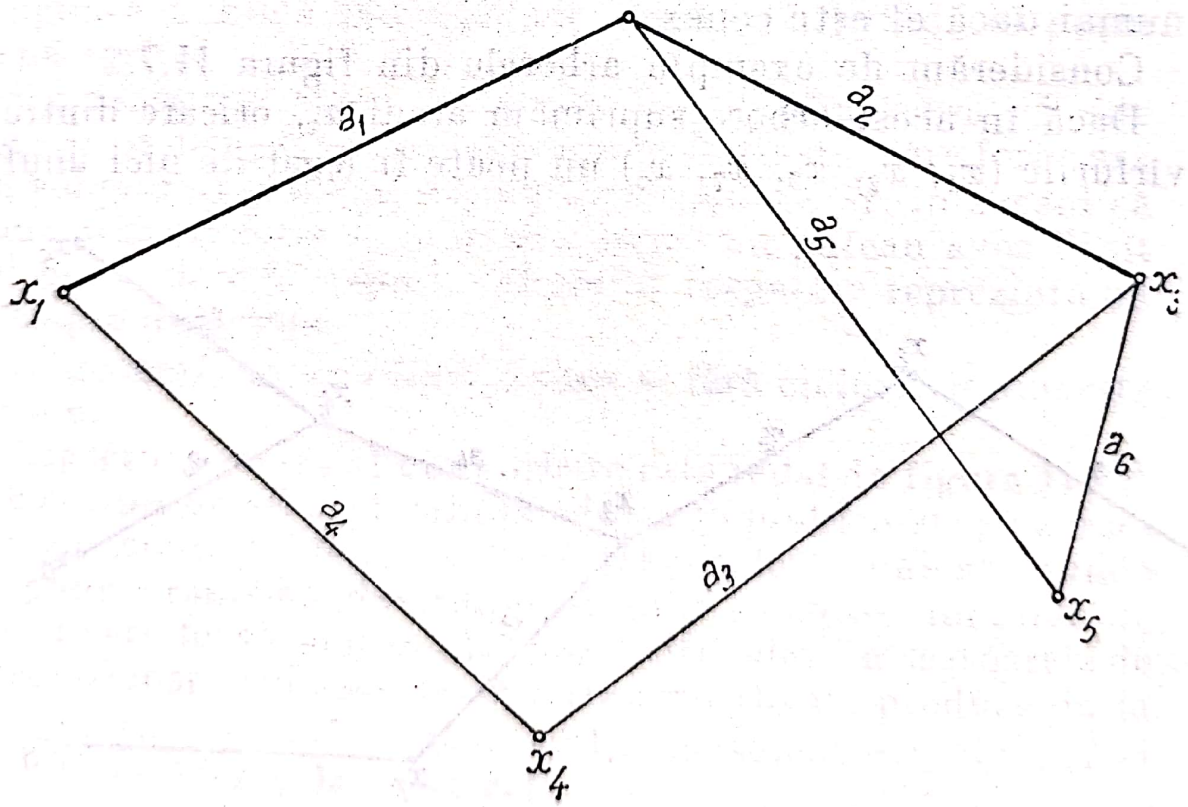


Fig. II.7.3

Scanned with OKEN Scanner

Din proprietăţile de mai sus deducem cu uşurinţă un algoritm simplu pentru găsirea unui arbore într-un graf conex.

2°. **Algoritm pentru determinarea unui arbore într-un graf conex.** Fie graful $G = (X,A)$:

a) Plecăm de la un arc arbitrar ales $a_{i_1} \in A$.

b) În graful G alegem un arc $a_{i_2} \in A$, care nu formează ciclu cu arcul a_{i_1}.

c) Alegem din nou un arc $a_{i_3} \in A$, care nu formează cicluri cu arcele a_{i_1} şi a_{i_2}.

Continuăm alegerea arcelor $a_{i_k} \in A$, în condiţiile arătate cît este posibil. Presupunem că arcul a_{i_p} este ultimul arc ales.

Succesiunea de arce $\{a_{i_2}, a_{i_1}, \ldots, a_{i_k}, \ldots, a_{i_p}\}$ formează un arbore în graful dat, deoarece graful parţial format cu arcele $\{a_{i_k}\}_{k=1,2,\ldots,p}$ este conex şi fără cicluri.

Oricare din arcele gradului G rămase în afara arborelui formează cicluri cu arcele arborelui.

3°. **Cicluri într-un graf şi legătura cu vectorii.** Reluăm acum unele definiţii legate de conceptul de graf.

Am văzut că noţiunea de graf pune în evidenţă o mulţime de vîrfuri X, o mulţime de arce A şi o lege de corespondenţă F între vîrfurile grafului.

În capitolul I au fost date unele definiţii utile pe care de această dată le reluăm şi le extindem.

Aşa, de exemplu, am definit ciclul într-un graf ca o succesiune de muchii astfel construită, încît o muchie m_i să fie legată prin una din extremităţile sale de muchia m_{i-1}, iar prin cealaltă extremitate, de muchia m_{i+1}, cu condiţia ca o muchie să figureze o singură dată în această succesiune, iar extremitatea finală a ultimei muchii să coincidă cu extremitatea iniţială a celei dintîi.

Putem defini un ciclu γ drept un vector pe mulţimea M a arcelor grafului G. Notînd componentele vectorului γ cu γ_i unde indicele i ia atîtea valori cîte arce are graful, componentele γ_i sînt egale cu:

Scanned with OKEN Scanner

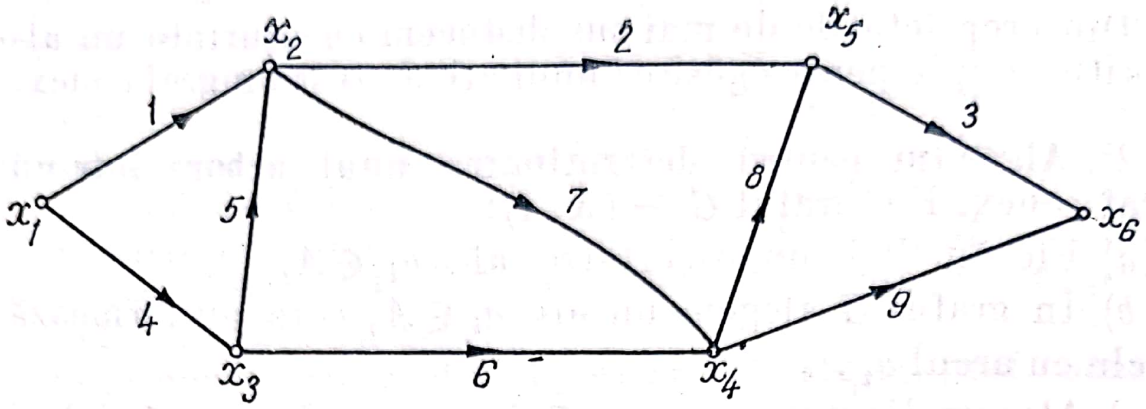


Fig. II.7.4

0, dacă arcul a_i nu face parte din ciclu,
+ 1, dacă arcul a_i face parte din ciclu și este orientat în sensul de parcurs,
— 1, dacă arcul a_i face parte din ciclu, dar este orientat în sens contrar sensului de parcurs.

În graful din figura II.7.4, vîrfurile x_2 x_5 x_4 x_3 definesc un ciclu {2, 8, 6, 5} care poate fi scris ca un vector de componente: (0, +1, 0, 0, +1, —1, 0, —1, 0).

Ținînd seama de posibilitatea reprezentării unui ciclu printr-un vector, putem defini dependența și independența liniară a ciclurilor.

Spunem că ciclurile $\gamma^{(1)}$, $\gamma^{(2)}$, ..., $\gamma^{(p)}$ sînt *dependente*, dacă există p, scalari, λ_1, λ_2, ..., λ_p nu toți nuli în același timp, astfel încît să avem satisfăcută relația:

$$\lambda_1 \gamma^{(1)} + \lambda_2 \gamma^{(2)} + \ldots + \lambda_p \gamma^{(p)} = 0.$$

Dimpotrivă, dacă aceeași relație este satisfăcută numai pentru valori λ_i, toate nule în același timp, spunem că ciclurile sînt *independente*.

Corespunzător bazei unui sistem de vectori, putem vorbi acum de baza ciclurilor.

Spunem că mulțimea de cicluri $\gamma^{(1)}$, $\gamma^{(2)}$, ..., $\gamma^{(r)}$ formează o bază a ciclurilor grafului dat, dacă oricare ar fi ciclul γ, poate fi exprimat ca o combinație liniară a acestora, adică,

$$\gamma = \lambda_1 \gamma^{(1)} + \lambda_2 \gamma^{(2)} + \ldots + \lambda_r \gamma^{(r)},$$

unde (λ_i) $(1 = 1, 2, \ldots,)$ sînt numere reale (scalari).

O bază a ciclurilor o obținem în modul următor:

a) Determinăm un arbore parțial G^* al lui G;

Scanned with OKEN Scanner

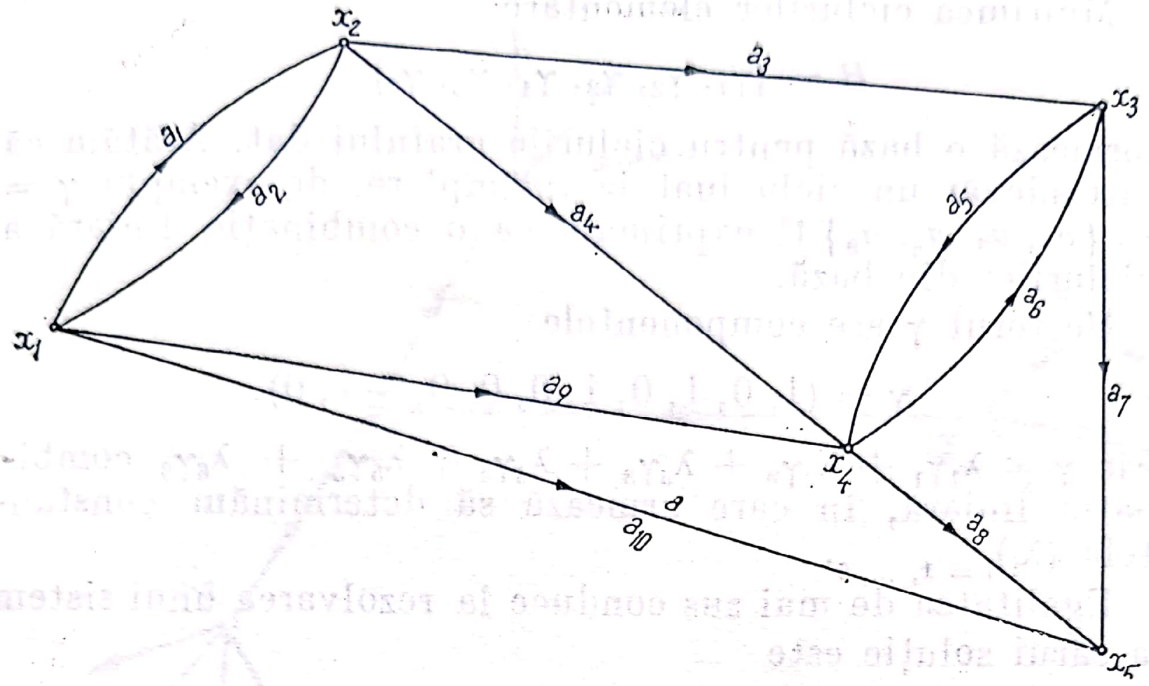


Fig. II.7.5

b) Orice arc al grafului G făcînd parte din graful parțial $G—G^*$, adăugat lui G^*, determină un ciclu elementar. Dacă graful are „n" vîrfuri în componența arborelui G^*, știm că intră $(n — 1)$ arce. Graful $G—G^*$ are deci $|A| — (n — 1)$ arce, unde am notat cu $|A|$ numărul de arce al grafului dat.

Deci numărul ciclurilor elementare pe care le conține un graf $G = (X, A)$ este dat de $|A| — (n — 1)$. Mulțimea ciclurilor elementare astfel obținute formează o bază.

Urmînd acest procedeu în graful din figura II.7.5, arborele G^* este format din arcele a_1, a_4, a_5, a_7, marcat cu linii roșii. Întrucît arborele conține patru arce, iar $|A| = 10$, înseamnă că în graful G există $|A| — (n — 1) = 10 — — 4 = 6$ cicluri elementare. Acestea sînt:

$$\gamma_1 = (a_1, a_2) = (1, —1, 0, 0, 0, 0, 0, 0, 0, 0)$$
$$\gamma_2 = (a_1, a_4, a_9) = (1, 0, 0, 1, 0, 0, 0, 0, —1, 0)$$
$$\gamma_3 = (a_3, a_5, a_4) = (0, 0, 1, —1, 1, 0, 0, 0, 0, 0)$$
$$\gamma_4 = (a_5, a_6) = (0, 0, 0, 0, 1, 1, 0, 0, 0, 0)$$
$$\gamma_5 = (a_5, a_7, a_9) = (0, 0, 0, 0, —1, 0, 1, —1, 0, 0)$$
$$\gamma_6 = (a_1, a_4, a_5, a_7, a_{10}) = (1, 0, 0, 1, —1, 0, 1, 0, 0, — 1)$$

Scanned with OKEN Scanner

Mulțimea ciclurilor elementare

$$B = \{\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6\}$$

formează o bază pentru ciclurile grafului dat. Arătăm că într-adevăr un ciclu luat la întîmplare, de exemplu $\gamma = \{a_1, a_3, a_5, a_9\}$ îl exprimăm ca o combinație liniară a ciclurilor din bază.

Vectorul γ are componentele

$$\gamma = (1, 0, 1, 0, 1, 0, 0, 0, -1, 0).$$

Fie $\gamma = \lambda_1\gamma_1 + \lambda_2\gamma_2 + \lambda_3\gamma_3 + \lambda_4\gamma_4 + \lambda_5\gamma_5 + \lambda_6\gamma_6$ combinația liniară, în care urmează să determinăm constantele $(\lambda_i)_{i=1,\ldots,6}$.

Egalitatea de mai sus conduce la rezolvarea unui sistem a cărui soluție este

$$\lambda_1 = \lambda_4 = \lambda_5 = \lambda_6 = 0$$

și

$$\lambda_2 = \lambda_3 = 1, \text{ adică}$$
$$\gamma = \gamma_2 + \gamma_3.$$

4°. **Arborescență.** Numim „arborescență" un graf fără circuite, în care există un vîrf (x_0) care nu are strămoși $(F_{x_0}^{-1} = \emptyset)$, iar toate celelalte vîrfuri ale grafului reprezintă extremitatea finală a unui arc și numai a unuia, cu alte cuvinte, din oricare alt vîrf al grafului, diferit de x_0, pleacă oricîte arce, dar nu sosește decît unul singur. De exemplu, graful din figura II.7.6 reprezintă o arborescență.

Vîrful x_0, în care nu ajunge nici un arc, se numește „rădăcina arborescenței".

Un exemplu referitor la noțiunea de arborescență îl constituie arborele genealogic.

În acest caz, arcele reprezintă relația „este fiul lui".

În anumite cazuri putem găsi într-un graf G un graf parțial G^*, care să fie o arborescență. De această dată obținem o „arborescență parțială" a grafului dat. Noțiunea de „arborescență parțială" a unui graf o întîlnim în probleme practice cum sînt cele legate de lingvistică, traduceri, precum și în matematicile pure.

Scanned with OKEN Scanner

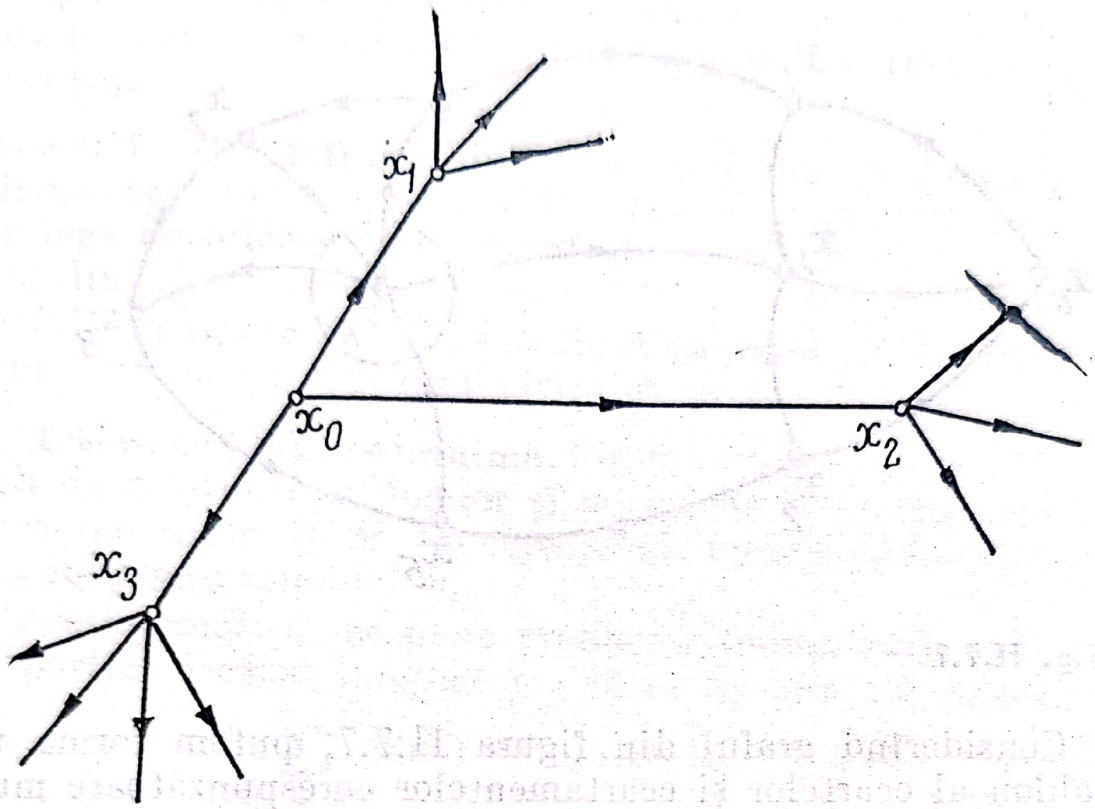


Fig. II.7.6

Ecart într-un graf

Se numește „*ecart*“ între două vîrfuri x_i și x_k ale unui graf lungimea celui mai scurt drum care duce de la x_i la x_k.

Dacă $x_i = x_k$ rezultă $d(x_i, x_k) = d(x_i, x_i) = 0$.

Dacă $x_k \in \hat{F}x_i$ rezultă $d(x_i, x_k) = \infty$.

Se numește „ecartamentul“ unui vîrf $x \in X$ valoarea

$$e(x) = \max_{y \in X} d(x, y).$$

Punctul căruia îi corespunde ecartamentul minim într-un graf se numește „centru“ al grafului, în timp ce punctul căruia îi corespunde ecartamentul maxim se numește punct „periferic“.

Dacă $\bar{x}$ este un centru al grafului, se numește „rază“, ecartamentul punctului $\bar{x}$.

Dacă graful nu admite un centru, zicem că raza grafului este infinită.

Scanned with OKEN Scanner

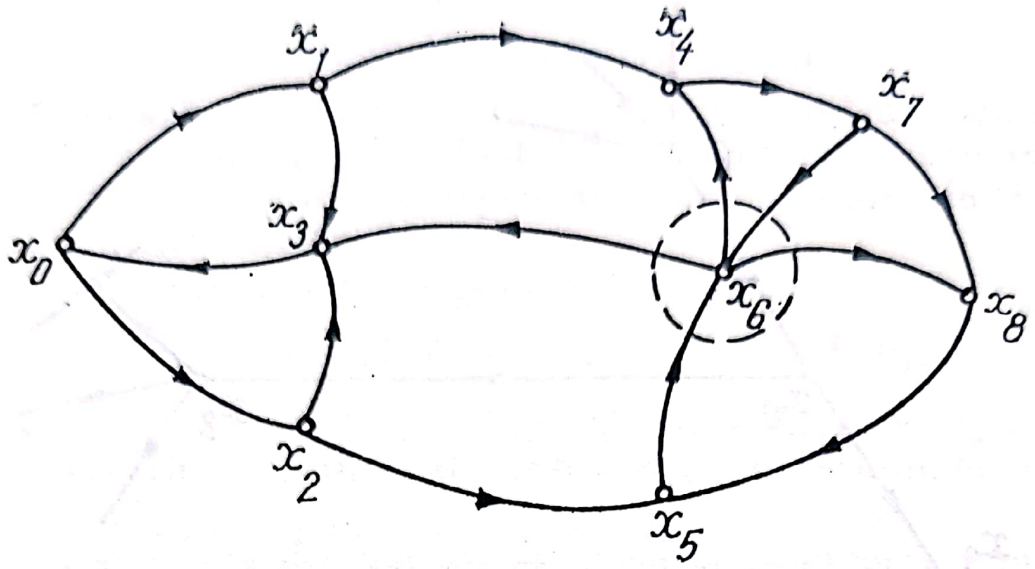


Fig. II.7.7

Considerînd graful din figura II.7.7, putem forma un tablou al ecartelor și ecartamentelor corespunzătoare mulțimii $X = \{x_i\}_{i=0,\ldots,8}$.

În acest scop formăm matricea ale cărei elemente ρ_{ij} reprezintă ecartul dintre vîrful x_i și x_j.

Pe ultima coloană a matricei, vom trece ecartamentul fiecărui vîrf x_i, adică elementul maxim situat pe linia vîrfului x_i.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
x_0	0	1	1	2	2	2	3	3	4	4
x_1	2	0	3	1	1	4	3	2	3	4
x_2	2	3	0	1	4	1	2	4	3	4
x_3	1	2	2	0	3	3	4	4	5	5
x_4	4	5	5	3	0	3	2	1	2	5
x_5	3	4	4	2	2	0	1	3	2	4
x_6	2	3	3	1	1	2	0	2	1	3
x_7	3	4	4	2	2	2	1	0	1	4
x_8	4	5	5	3	3	1	2	4	0	5

Vîrful de ecartament minim este vîrful x_6 cu $e(x_6) = 3$, deci el constituie singurul centru al grafului dat. Raza

grafului este egală cu 3 unităţi. Vîrfurile x_3, x_4 şi x_8 sînt vîrfuri periferice, deoarece lor le corespunde ecartamentul maxim (5).

Observaţii: Din definiţia dată centrului unui graf rezultă că vîrful care are proprietatea de a fi centru al grafului se poate lega de oricare vîrf al grafului printr-un drum care pleacă din el.

Graful din figura II.7.8 reprezintă un exemplu de arborescenţă, avînd ca rădăcină vîrful $x_3 = x_c$.

5°. **Arbore de lungime minimă.** Fie dat graful $G = (X, M)$, definit de mulţimea vîrfurilor şi mulţimea muchiilor sale. Fiecărei muchii $m_i \in M$, îi corespunde un număr $l(m_i) \geqslant 0$, numit lungimea muchiei m_i.

În aceste condiţii, se pune problema determinării arborelui parţial a cărui lungime totală să fie minimă, adică:

$$\sum_{m_i \in M^*} l(m_i) = \text{minim},$$

unde M^* este mulţimea muchiilor care intră în componenţa arborelui.

Exemple de acest fel se întîlnesc în mod frecvent în telecomunicaţii, unde trebuie să se găsească cea mai mică lun-

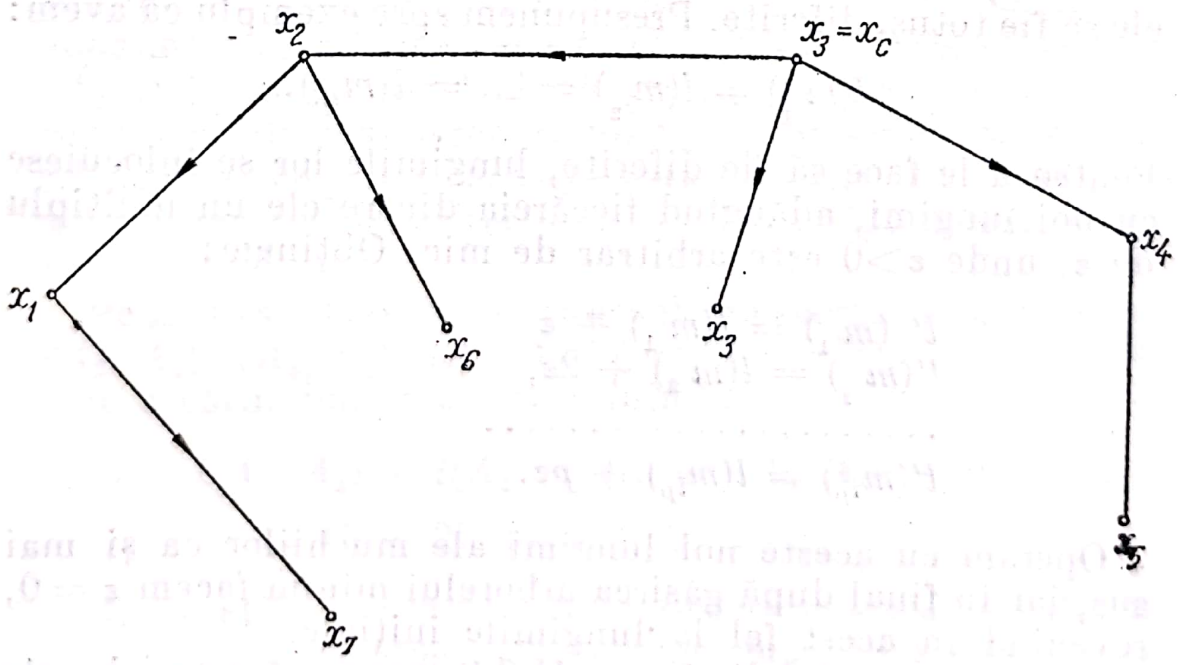


Fig. II.7.8

Scanned with OKEN Scanner

gime posibilă de cablu care leagă între ele „n" localități, în probleme de aprovizionare, etc.

În acest caz, localitățile reprezintă vîrfurile x_i ale grafului, iar lungimile arcelor $l(x_1, x_j)$ distanțele în kilometri între localitățile x_i și localitățile x_j. Evident că rețeaua de cablu trebuie să fie conexă și fără circuite, pentru a asigura legătura dintre toate localitățile.

Pentru găsirea arborelui de lungime minimă au fost elaborate mai multe algoritme.

Printre acestea, menționăm algoritmele lui Kruskal și Sollin.

6°. **Algoritmul lui Kruskal** constă în alegerea celor mai mici muchii, astfel încît ele să nu formeze cicluri cu muchiile anterior alese. În felul acesta, vom găsi o mulțime de $(n-1)$ muchii, $M^*_{n-1} = \{m^*_1, \ldots, m^*_{n-1}\}$, care formează arborele minim.

Graful $G^* = (X, M^*_{n-1})$, obținut prin selecționarea muchiilor de lungime minimă în modul arătat, constituie un arbore a cărui lungime totală este minimă.

Observații: Dacă graful conține mai multe muchii care au aceeași lungime, acestea le putem modifica în așa fel ca ele să fie totuși diferite. Presupunem spre exemplu că avem:

$$l(m_{i_1}) = l(m_{i_2}) = \ldots = l(m_{i_p}).$$

Pentru a le face să fie diferite, lungimile lor se înlocuiesc cu noi lungimi, adăugînd fiecăreia dintre ele un multiplu de ε, unde $\varepsilon > 0$ este arbitrar de mic. Obținem:

$$\begin{aligned} l'(m_{i_1}) &= l(m_{i_1}) + \varepsilon \\ l'(m_{i_2}) &= l(m_{i_2}) + 2\varepsilon, \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ l'(m_{i_p}) &= l(m_{i_p}) + p\varepsilon. \end{aligned}$$

Operăm cu aceste noi lungimi ale muchiilor ca și mai sus, iar în final după găsirea arborelui minim facem $\varepsilon = 0$, revenind în acest fel la lungimile inițiale.

Arătăm pe graful din figura II.7.9 cum se găsește arborele de lungime minimă folosind algoritmul descris.

Scanned with OKEN Scanner

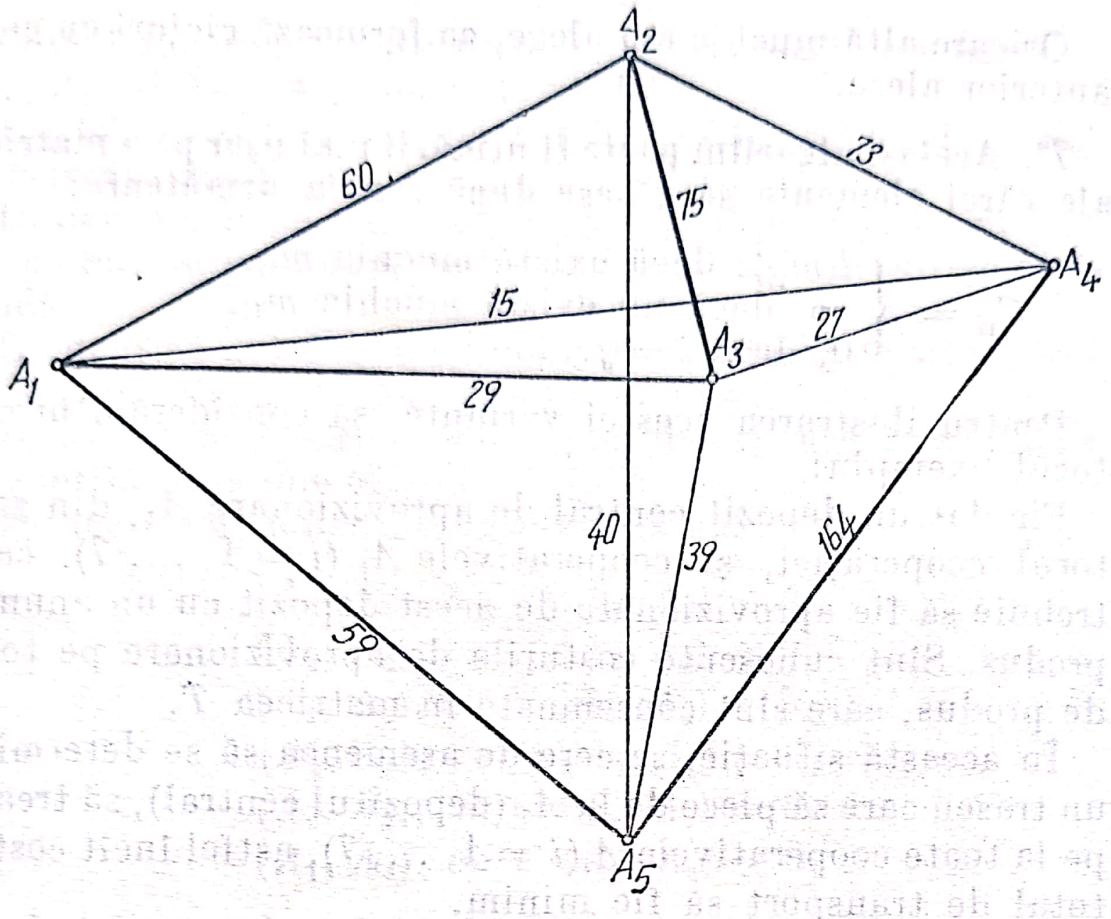


Fig. II.7.9

Conform algoritmului, plecăm de la muchia de lungime minimă (A_1, A_4) în continuarea căreia luăm muchia de lungime minimă imediat superioară celei alese, și care este (A_4, A_3). Dintre muchiile rămase, o luăm pe aceea care are lungimea cît mai mică și nu formează ciclu cu muchiile (A_1, A_4) și (A_4, A_3); aceasta este (A_3, A_5). În sfîrșit, vom alege după același criteriu muchia (A_5, A_3), obținînd arborele minim.

Pe graf este trasată cu linie subțire succesiunea de muchii (A_1, A_4), (A_4, A_3), (A_3, A_5), (A_5, A_2), care formează arborele a cărui lungime este minimă:

$$l(Y) = l(A_1, A_4) + l(A_4 + A_3) + l(A_3, A_5) + l(A_5, A_2) =$$
$$= 15 + 27 + 39 + 40 = 121.$$

Observații: La cea de-a treia alegere muchia (A_1, A_3) deși era imediat următoare ca mărime, muchiilor alese, nu s-a luat în considerație, deoarece ea forma cu acestea un ciclu.

Scanned with OKEN Scanner

Oricare altă muchie am alege, ea formează cicluri cu cele anterior alese.

7°. **Același algoritm poate fi urmărit mai ușor pe o matrice ale cărei elemente se găsesc după regula următoare:**

$$C_{ij} = \begin{cases} l(m_{ij}), \text{ dacă există muchia } m_{ij}, \\ \infty, \text{ dacă nu există muchia } m_{ij}, \\ 0, \text{ dacă } i = j. \end{cases}$$

Pentru ilustrarea acestei variante, să considerăm următorul exemplu:

Fie dat un depozit central de aprovizionare A_1, din sectorul cooperației, și cooperativele A_i ($i = 1, \ldots, 7$), care trebuie să fie aprovizionate de acest depozit cu un anumit produs. Sînt cunoscute costurile de aprovizionare pe tona de produs, care sînt consemnate în matricea T.

În această situație, se cere de asemenea să se determine un traseu care să plece de la A_1 (depozitul central), să treacă pe la toate cooperativele $A_i (i = 1, \ldots, 7)$, astfel încît costul total de transport să fie minim.

$T =$

	A_1	A_2	A_3	A_4	A_5	A_6	A_7
A_1	0	10	12	8^{V}	14	15	14
A_2	10	0	∞	∞	13	11	6^{III}
A_3	12	∞	0	7^{IV}	6^{III}	5^{II}	8^{V}
A_4	8	∞	7	0	5^{II}	∞	15
A_5	14	13	6	5	0	7^{IV}	12
A_6	15	11	5	∞	7	0	4^{I}
A_7	14	6	8	15	12	4	0

În matricea T se alege muchia cu costul cel mai mic. În cazul nostru, costul cel mai mic, de patru unități, corespunde muchiei (A_6, A_7); marcăm acest cost în tabela T cu I.

Costul imediat superior lui (4) este (5), corespunzător muchiilor (A_3, A_6) și (A_4, A_5); marcate în tabel cu II.

Costul superior lui (5) este (6), corespunzător muchiilor (A_3, A_5) și (A_2, A_7); marcate cu III.

Scanned with OKEN Scanner

Costul superior lui (6) este (7), corespunzător muchiilor (A_3, A_4) și (A_5, A_6); marcat cu IV.

La fel, costul (8) de pe muchiile (A_1, A_4) și (A_3, A_7) îl marcăm cu V.

Dintre muchiile marcate, vom reține pe acelea care nu fac cicluri cu precedentele lor și cu costurile în ordine crescîndă.

Evident că numărul de muchii alese este egal cu numărul de vîrfuri ale grafului, mai puțin unu.

În exemplul nostru avînd șapte vîrfuri, arborele trebuie să conțină șase muchii.

În acest fel, muchiile arborelui sînt:

$$(A_6, A_7), \text{ cu } C\ (A_6, A_7) = 4,$$
$$(A_3, A_6), \text{ cu } C\ (A_3, A_6) = 5,$$
$$(A_4, A_5), \text{ cu } C\ (A_4, A_5) = 5,$$
$$(A_3, A_5), \text{ cu } C\ (A_3, A_5) = 6,$$
$$(A_2, A_7), \text{ cu } C\ (A_2, A_7) = 6.$$
$$(A_1, A_4), \text{ cu } C\ (A_1, A_4) = 8,$$

iar lungimea minimă a arborelui este:

$$C(Y) = 8 + 5 + 6 + 7 + 4 + 6 = 36$$

Arborele de lungime minimă este redat în figura II.7.10.

Observații: 1. În alegerea muchiilor au fost omise muchiile (A_3, A_4) și (A_5, A_6) cu cost (7), deoarece ele făceau cicluri cu muchiile anterior alese. Astfel, s-a trecut la muchia cu costul imediat superior (8).

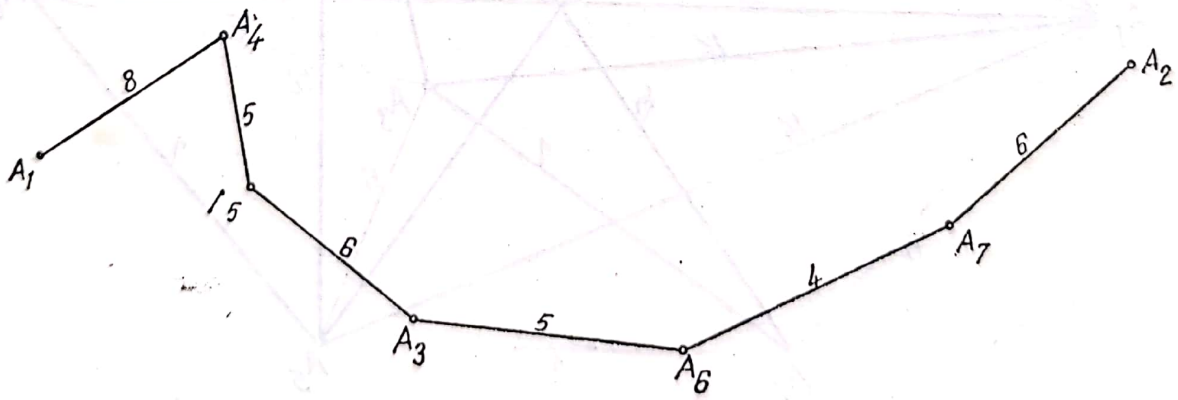


Fig. II.7.10

2. Matricea corespunzătoare grafului fiind simetrică, se lucrează numai cu elementele situate de o parte sau de alta a diagonalei principale. Noi ne-am fixat asupra celor situate deasupra diagonalei principale.

3. Metoda stabilirii arborelui minim cu ajutorul matricei costurilor poate porni și de la eliminarea costurilor mari, cu respectarea celor de la punctul 1.

4. Metoda costurilor poate fi înlocuită cu o altă metodă prin care elementele considerate în ordinea mărimii se înlocuiesc cu unitatea. În momentul în care obținem cel puțin $(n - 1)$ elemente egale cu 1, restul de elemente le înlocuim cu zero.

În felul acesta, elementele de deasupra diagonalei principale din tabelul T sînt formate din 0 și 1. Completînd prin simetrie restul elementelor matricei, obținem o matrice booleană.

Din această matrice reținem atîtea elemente 1 cîte sînt necesare pentru a construi cu muchiile corespunzătoare lor arborele a cărui lungime este evident minimă.

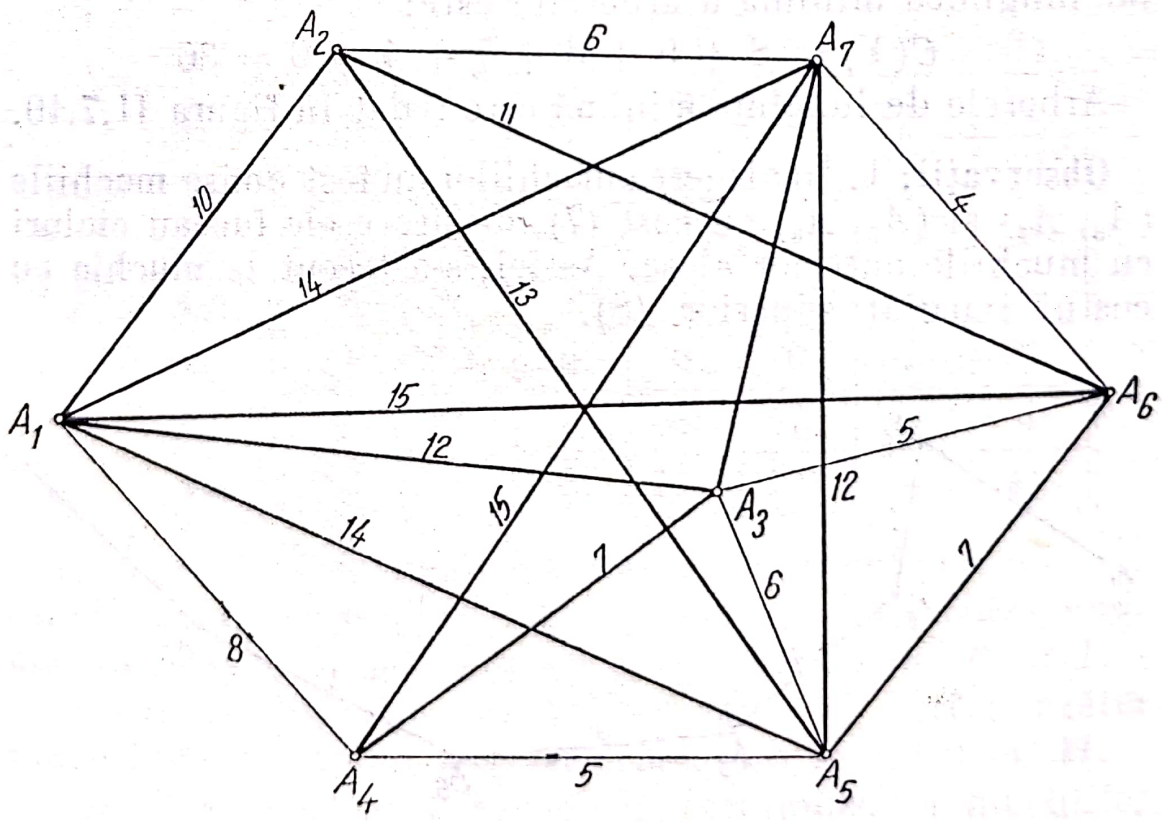


Fig. II.7.11

Scanned with OKEN Scanner

Să aplicăm algoritmul lui Sollin pe exemplul din figura II.7.11

Cel mai apropiat vîrf de A_1 este A_4, la distanța de 8 unități, față de A_4 este A_5, față de A_5 este A_3, față de A_3 este A_6, față de A_6 este A_7 și față de A_7 este A_2.

Observație: Constatăm că pentru un graf cu număr mic de vîrfuri algoritmul lui Sollin este foarte simplu în comparație cu algoritmul lui Kruskal.

Scanned with OKEN Scanner

CAPITOLUL

3

MODELE DE REȚELE APLICATE ÎN ECONOMIE

A. Rețele de transport

§ 1. INTRODUCERE

Datorită importanței pe care o au modelele de rețele în analiza proceselor complexe ale producției, consacrăm acest capitol studierii unor modele care intervin în diferite sectoare de activitate economică.

Teoria rețelelor se referă la probleme generale cu aplicații în domenii foarte variate, cum sînt cele legate de transporturile rutiere, navale, aeriene, canalizări, planificare, control și conducere, repartizări de sarcini etc.

Exemplul 1. Considerăm porturile maritime $x_1, x_2, \ldots, x_k$ $x_{k+1}, \ldots, x_n$ și un același produs disponibil porturilor $x_{i\,(i=1,2,\ldots,k)}$ în cantități $a_{i\,(1=1,2,\ldots,k)}$ și același produs necesar porturilor $x_{j\,(j=k+1,\ldots,n)}$ în cantități $b_{j\,(j=k+1,\ldots,n)}$.

Cu ajutorul acestor date putem construi un graf în care porturile $x_{i\,(i=1,2,\ldots,k)}$ și $x_{j\,(j=k+1,\ldots,n)}$ vor fi reprezentate prin puncte, iar căile maritime prin arce (x_i, x_j). Într-o asemenea problemă, se poate cere volumul maxim de produse ce poate fi transportat de la porturile, $x_{i\,(i=1,\ldots,k)}$ la porturile $x_{j\,(j=k+1,\ldots,n)}$, abstracție făcînd de costurile de transport.

Exemplul 2. a) Fie de îndeplinit n sarcini pe m mașini unde $n \geqslant m$. Admitem că fiecare mașină poate efectua în mod satisfăcător numai o parte din aceste sarcini. În aceste condiții, se pune problema repartizării optime a sarcinilor pe mașini.

b) Presupunem însă că fiecare mașină poate efectua toate sarcinile, dar randamentul mașinilor variază în funcție de

Scanned with OKEN Scanner

sarcini. Se cere o repartizare a sarcinilor pe mașini, astfel încît eficiența economică să fie maximă.

c) Admitem că se cunoaște succesiunea sarcinilor și că mașinile pot efectua toate sarcinile. Se cere o repartizare, astfel ca un număr minim de mașini să lucreze într-un interval de timp determinat.

În astfel de probleme, dacă notăm cu $x_{i\,(i=1,\ldots,n)}$ sarcinile și cu $x_{j\,(j=n+1,\ldots,n+m)}$ mașinile, putem construi un graf ale cărui vîrfuri vor fi punctele $x_{i(i=1,\ldots,n)}$ și $x_{j\,(j=n+1,\ldots,n+m)}$ și ale cărui arce (x_i, x_j) vor arăta posibilitatea executării sarcinii x_i pe mașina x_j.

Forma cea mai generală sub care poate să se prezinte o rețea este cea din figura III.A.1.1.

Din analiza acestei figuri rezultă că în general o rețea este un graf finit fără bucle, în care mulțimea X a vîrfurilor sale se împarte în trei submulțimi distincte:

x_1 formată din vîrfuri $x_k^{(j)}$ cu $F^{-1}_{x_k}(j) = \emptyset$; $j = 1,2, \ldots, q$,

x_2 formată din vîrfuri $x_l^{(i)}$ cu $F_{x_l}(j) = \emptyset$; $j = 1,2, \ldots, p$,

x_3 formată din puncte $x_i^{(j)}$ cu $F_{x_i}(j) \neq \emptyset$ și $F^{-1}_{x_i}(j) \neq \emptyset$.

Modelele de rețele care intervin în aplicațiile practice se pot însă simplifica, în sensul că mulțimea x_1 se reduce

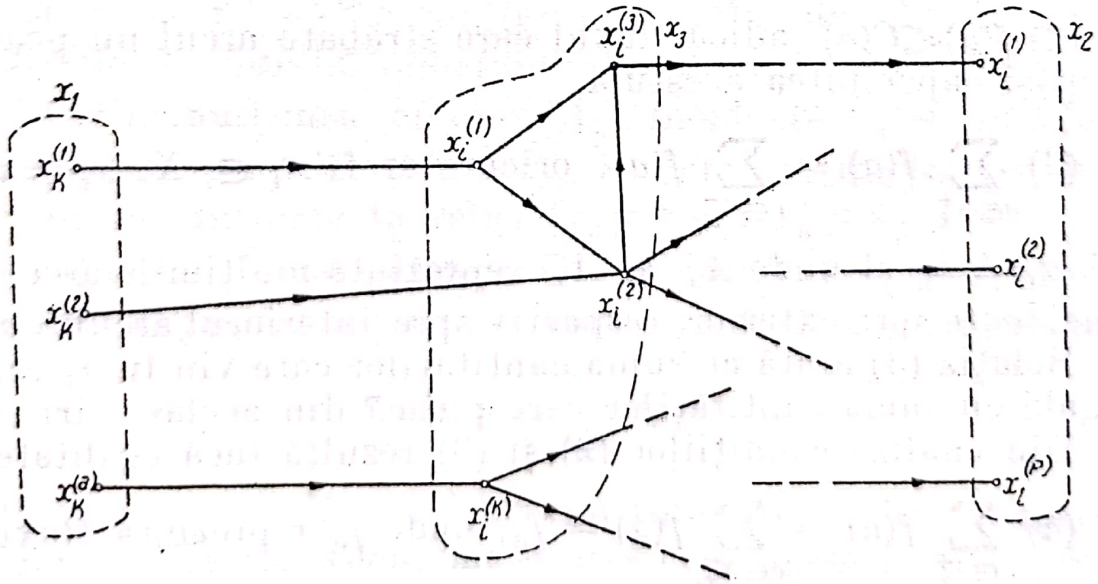


Fig. III.A.1.1

la un singur punct x_0, numit intrare în rețea; x_2 se reduce la un singur punct x_n, numit ieșirea din rețea, iar mulțimea x_3 va fi formată din toate celelalte vîrfuri ale grafului, $x_1, \ldots, x_{n-1}$.

Dacă în rețeaua astfel simplificată atașăm arcelor numere nenegative $C(a)$ numite capacități, obținem un model de „rețea de transport".

Evident că numerele $C(a)$ sînt finite.

Cu alte cuvinte putem defini o „rețea de transport" ca fiind un graf finit fără bucle avînd o intrare și numai una, o ieșire și numai una și pe care este definită o funcție de arc $C(a)$ nenegativă și cu valori finite, numită capacitatea arcului „a".

§ 2. FLUX ÎNTR-O REȚEA DE TRANSPORT

Noțiunea de „flux" este specifică conceptului de rețea.

Fie graful $G = (X, A)$, fluxul corespunzător unui arc $a \in A$, notat cu $f(a)$, reprezintă cantitatea de materie ce străbate arcul respectiv și care satisface condițiile:

(1) $f(a) \geqslant 0$, relație care ne spune că fluxul nu poate fi un număr negativ (evident că o cantitate transportată nu poate fi exprimată printr-un număr negativ);

(2) $f(a) \leqslant C(a)$, adică fluxul care străbate arcul nu poate depăși capacitatea acestuia.

(3) $\sum_{a \in A^+_{x_i}} f(a) = \sum_{a \in A^-_{x_i}} f(a)$, oricare ar fi $x_i \in X$, $x_i \neq x_0$ și $x_i \neq x_n$ și unde $A^+_{x_i}$ și $A^-_{x_i}$ reprezintă mulțimile arcelor incidente spre exterior, respectiv spre interiorul arcului x_i.

Relația (3) arată că suma cantităților care vin în x_i este egală cu suma cantităților care pleacă din același vîrf x_i.

Din analiza condițiilor (2) și (3) rezultă încă egalitatea

(4) $\sum_{a \in A^+_{x_i}} f(a) = \sum_{a \in A^-_{x_i}} f(a) = f_n$, unde f_n reprezintă fluxul în vîrful x_n (ieșirea din rețea).

Scanned with OKEN Scanner

Relația (4) arată că întreaga cantitate care pleacă din x_0 este egală cu cea care ajunge în x_n, ceea ce indică că nu au loc pierderi în rețea.

Problema de bază referitoare la rețelele de transport este aceea a determinării — după cerințele practice — a fluxului maxim sau minim.

Astfel, în problemele de fizică mai în toate cazurile se cere determinarea unui flux de la o sursă S, la o destinație D; în rețelele de comunicație adesea interesează fluxul de informație măsurat în secunde; în rețelele de trafic interesează cantitatea totală de materie ce poate fi transportată.

Din exemplele enumerate rezultă că problema fluxului maxim are sens numai în cazul în care arcele rețelei sînt limitate superior de anumite capacități $C(a)$, care exprimă posibilitatea maximă de transport pe arcul respectiv.

Înainte de a trece la găsirea unor metode care să ne permită determinarea fluxului maxim într-o rețea, este necesar să dăm unele definiții.

Definiția 1: Spunem că un arc este „*saturat*“ dacă fluxul care-l străbate este egal cu capacitatea sa, adică:

$$f(a) = C(a), \text{ pentru } a \in A.$$

Definiția 2: Un flux este „*complet*“ dacă oricare ar fi drumul de la x_0 la x_n acesta conține cel puțin un arc saturat.

Definiția 3: Numim „*tăietură sau secțiune*“ în rețeaua $G = (X, F)$, mulțimea arcelor A_y^- incidente spre interior unei submulțimi de vîrfuri $Y \subset X$, care conține ieșirea, dar nu conține intrarea în rețea ($x_0 \notin Y$) și $x_n \in Y$).

Definiția 4: „Capacitatea unei tăieturi“ A_y^- este egală cu suma capacităților arcelor care formează tăietura, adică

$$C(A_y^-) = \sum_{a \in A_y^-} C(a).$$

Pentru găsirea fluxului maxim într-o rețea de transport se cunosc diverse algoritme dintre care unele le vom prezenta în continuare.

Scanned with OKEN Scanner

§ 3. ALGORITME PENTRU DETERMINAREA FLUXULUI MAXIM (MINIM) ÎNTR-O REȚEA DE TRANSPORT

1°. **Algoritmul Ford-Fullkerson.** Acest algoritm pornește de la un flux inițial, arbitrar ales, care respectă condițiile impuse acestuia. Ținînd seama de scopul urmărit, construirea fluxului inițial trebuie făcută de așa manieră, încît el să satureze cît mai multe arce.

În pasul imediat următor se caută un flux complet, în sensul definiției date mai înainte.

După parcurgerea acestor două etape se verifică dacă fluxul găsit este sau nu maxim, folosind un procedeu de marcare a vîrfurilor rețelei.

Fiecare vîrf al rețelei va fi marcat cu două numere, dintre care primul să reprezinte cantitatea *ce mai poate sosi* în vîrful $x_i \in X$, iar al doilea să arate vîrful de la care provine această cantitate.

Vîrfului x_0 (intrarea în rețea) i se atașează perechea de numere $(\infty, 0)$, deoarece în x_0 este posibil să sosească orice cantitate de materie.

Dacă x_i este un vîrf marcat cu perechea de numere (b_i, k) (ceea ce înseamnă că în vîrful x_i mai putem aduce b_i unități din vîrful x_k), atunci:

a) marcăm vîrful x_j cu perechea de numere

$$[\min \{b_i, C(a_{ij}) - f(a_{ij})\}, i],$$

dacă arcul (x_i, x_j) este orientat în sensul de parcurs de la x_0 la x_n;

b) marcăm vîrful x_j, cu perechea de numere

$$[\min \{b_i, f(a_{ij})\}, i],$$

dacă arcul (x_i, x_j) este orientat în sens invers sensului de parcurs de la x_0 la x_n.

Dacă prin acest procedeu am reușit să marcăm vîrful x_n (ieșirea din rețea), atunci există un drum nesaturat.

De-a lungul acestui drum vom transporta cel mai mic număr de unități dat de primii indici atașați vîrfurilor marcate; această alegere este justificată de faptul că primul indice arată posibilitatea maximă de transport pe arcul

Scanned with OKEN Scanner

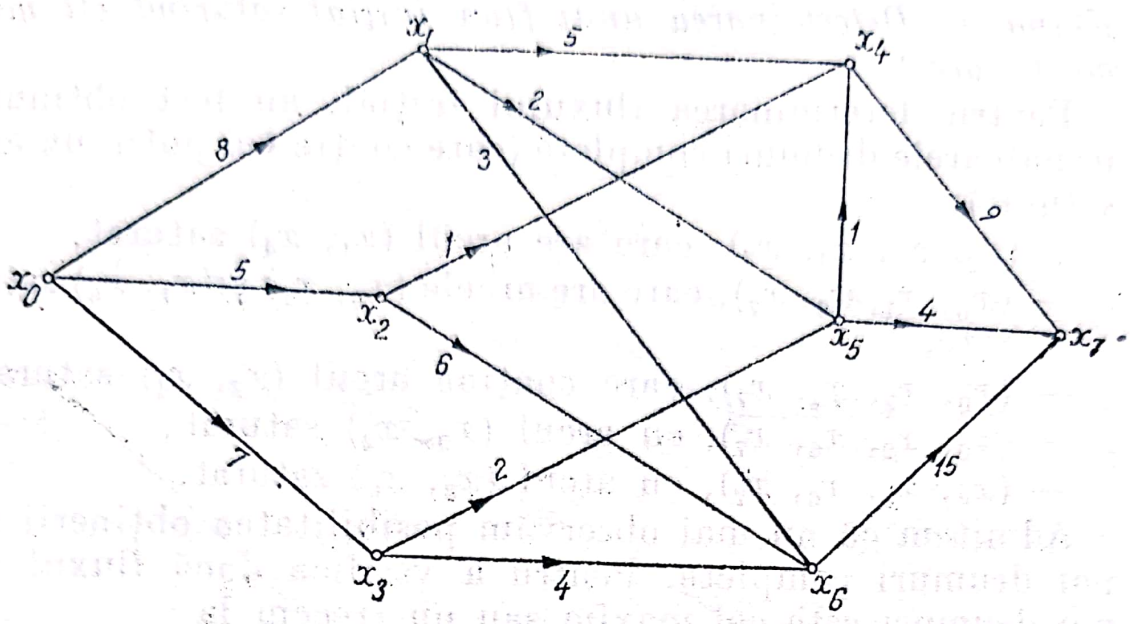


Fig. III.A.3.1

respectiv. Evident că dacă s-ar lua un indice corespunzător unei cantități mai mari decît cea aleasă, ar însemna că fluxul pe anumite arce să depășească capacitățile corespunzătoare.

Exemplul 1. Să exemplificăm cele relatate, considerînd rețeaua de transport din figura III.A.3.1.

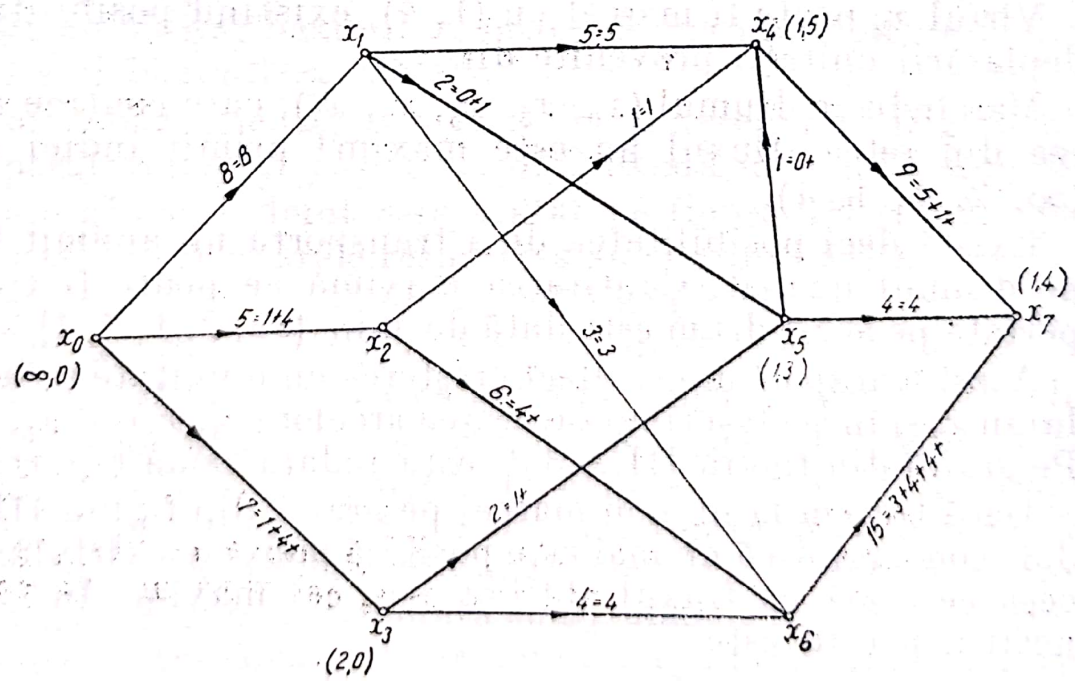


Fig. III.A.3.2

Etapa I. Determinarea unui flux inițial saturînd cît mai multe arce:

Pentru determinarea fluxului inițial, au fost obținute următoarele drumuri complete (care conțin cel puțin un arc saturat):

— (x_0, x_1, x_4, x_7), care are arcul (x_1, x_4) saturat,

— (x_0, x_1, x_6, x_7), care are arcele (x_0, x_1) și (x_1, x_6) saturate,

— (x_0, x_2, x_4, x_7), care conține arcul (x_2, x_4) saturat,

— (x_0, x_2, x_6, x_7), cu arcul (x_0, x_2) saturat;

— (x_0, x_3, x_6, x_7), cu arcul (x_3, x_6) saturat.

Admitem că nu mai observăm posibilitatea obținerii de noi drumuri complete. Pentru a verifica dacă fluxul de noi drumuri este cel maxim sau nu trecem la

Etapa a II-a. Marcarea vîrfurilor grafului.

Pe x_0 îl marcăm cu $(\infty, 0)$.

Pe x_3 îl marcăm cu $(2, 0)$, deoarece pe arcul (x_0, x_3), nesaturat, există disponibile două unități.

Pe x_5 îl putem marca cu $(1, 3)$, avînd posibilitatea să deplasăm o unitate din x_3.

Pe x_4 îl marcăm cu $(1, 5)$ întrucît pe arcul (x_5, x_4) poate fi deplasată o unitate provenită din x_5.

Vîrful x_7 poate fi marcat cu $(1, 4)$, existînd posibilitatea deplasării unității provenite din x_4.

Marcîndu-se drumul $(x_0, x_3, x_5, x_4, x_7)$, care conține ieșirea din rețea, fluxul nu este maxim; primii indici sînt $(\infty, 2, 1, 1, 1)$.

Există deci posibilitatea de a transporta un anumit flux pe drumul marcat. Cantitatea maximă ce poate fi transportată pe acest drum este dată de $\min \{\infty, 2, 1, 1, 1\} = 1$.

Acest transport are ca efect creșterea cu o unitate a fluxului în x_7 și în același timp saturarea arcelor (x_3, x_5) și (x_5, x_4). Pe graful din figura III.A.3.3 este redată noua repartiție.

Dacă trecem la un nou marcaj pe graful din figura III.A.3.3, constatăm că nu mai este posibilă marcarea vîrfului x_7, ceea ce arată că fluxul obținut este cel maxim. În cazul nostru, acesta este

$$f_{max} = 7 + 1 + 11 = 19 \text{ unități.}$$

Scanned with OKEN Scanner

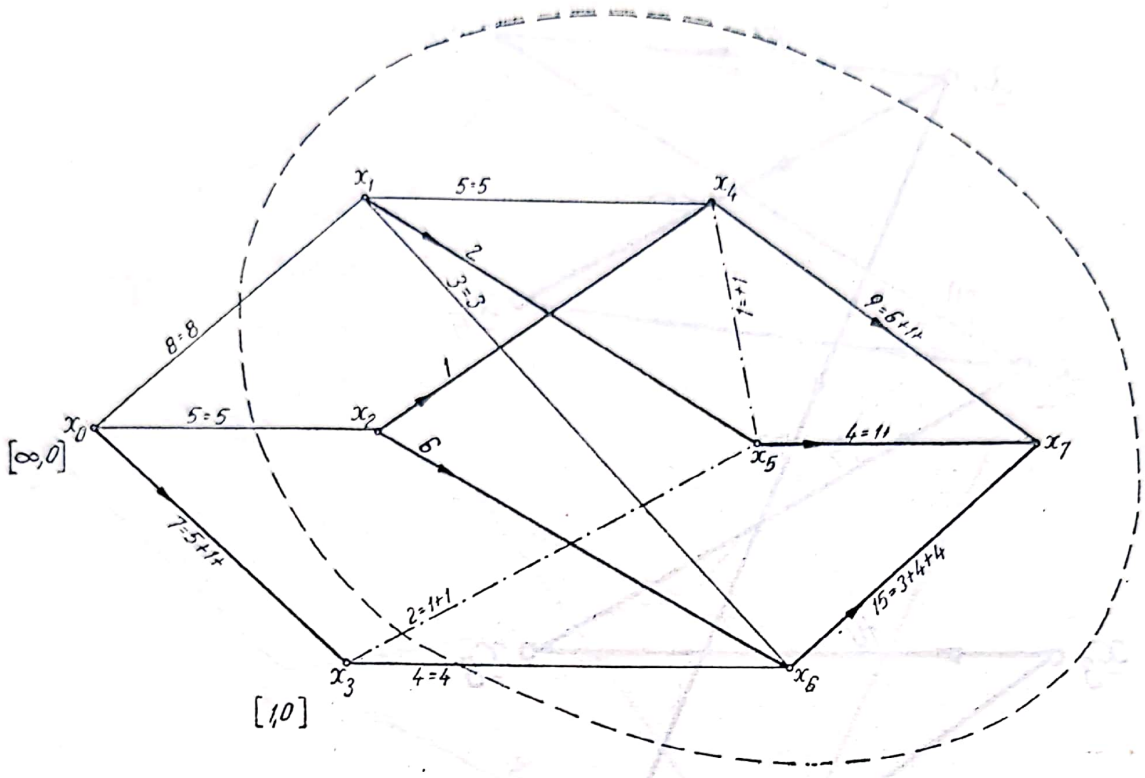


Fig. III.A.3.3

Exemplul 2. Fie trei bazine x_1, x_2, x_3 cu apă necesară irigării unor terenuri cultivate cu legume aparţinînd la patru cooperative agricole de producţie x_4, x_5, x_6, x_7.

Bazinul x_1 dispune de o rezervă de apă, capabilă de un debit de 45 l/s; bazinul x_2 poate să dea un debit de numai 25 l/s şi bazinul x_3, 20 l/s.

Între bazinele x_i, $i = 1, 2, 3$, şi terenurile irigate ale cooperativelor x_j, $j = 4, 5, 6, 7$ există deja un sistem de canale al căror debit este arătat pe figura III.A.3.4. Specialiştii din sectoarele respective au stabilit că C.A.P. x_4 ar avea nevoie de un debit de 30 l/s, C.A.P. x_5, de 10 l/s, C.A.P. x_6, de 20 l/s şi C.A.P. x_7, de 30 l/s.

Se cere să se stabilească cel mai bun mod posibil de alimentare cu apă şi să se determine dacă este cazul să se stabilească noi construcţii de canale suplimentare. Dacă grafului din figura III.A.3.4, care reprezintă canalizările existente cu debitele corespunzătoare, îi adăugăm o intrare notată cu x_0 şi o ieşire notată cu x_8, ambele fictive, se obţine o reţea de transport. Pe arcele (x_0, x_1), (x_0, x_2) şi (x_0, x_3) vom trece disponibilul de apă din bazinele x_1, x_2, x_3; de

Scanned with OKEN Scanner

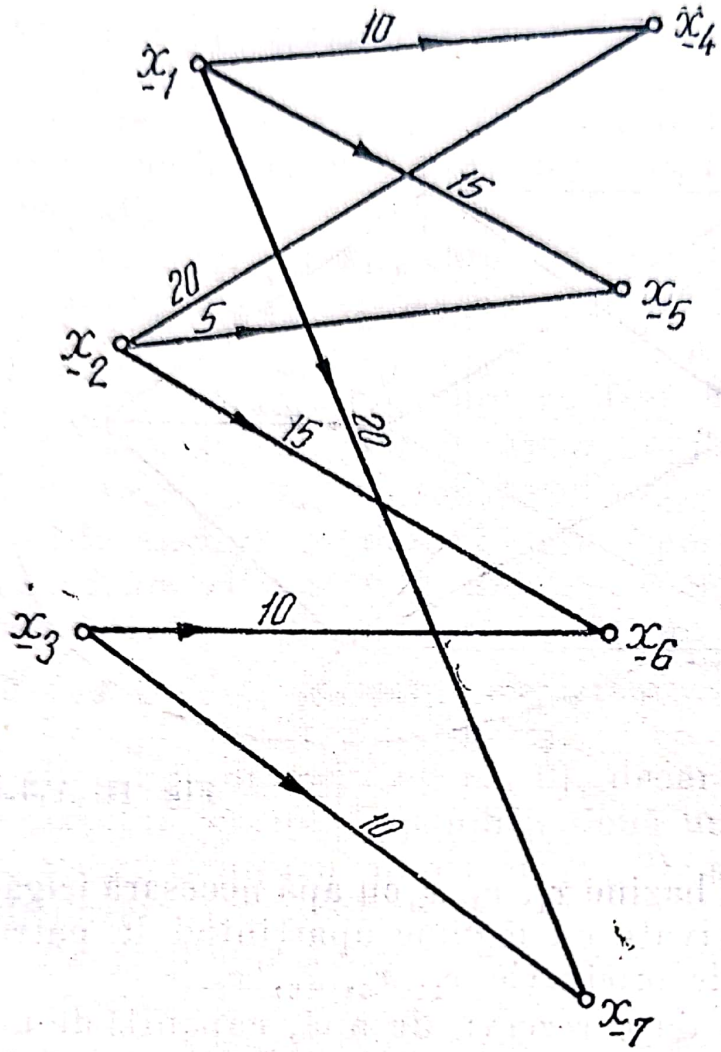


Fig. III.A.3.4

asemenea, pe arcele (x_4, x_8), (x_5, x_8), (x_6, x_8) și (x_7, x_8) se va trece necesarul de apă al cooperativelor agricole de producție x_4, x_5, x_6, x_7.

Cu aceste precizări obținem rețeaua de transport din figura III.A.3.5.

Găsirea debitului cel mai bun revine la găsirea fluxului maxim din rețeaua de transport din figura III.A.3.4. Pentru rezolvare vom folosi același algoritm.

Etapa 1. Construirea fluxului inițial $\varphi_0 = 20+10+20+$ $+30= 80$ este redată în figura III.A.3.6.

Deoarece vîrful x_8 (ieșirea din rețea) a fost marcat, există posibilitatea îmbunătățirii fluxului pe traseul

Scanned with OKEN Scanner

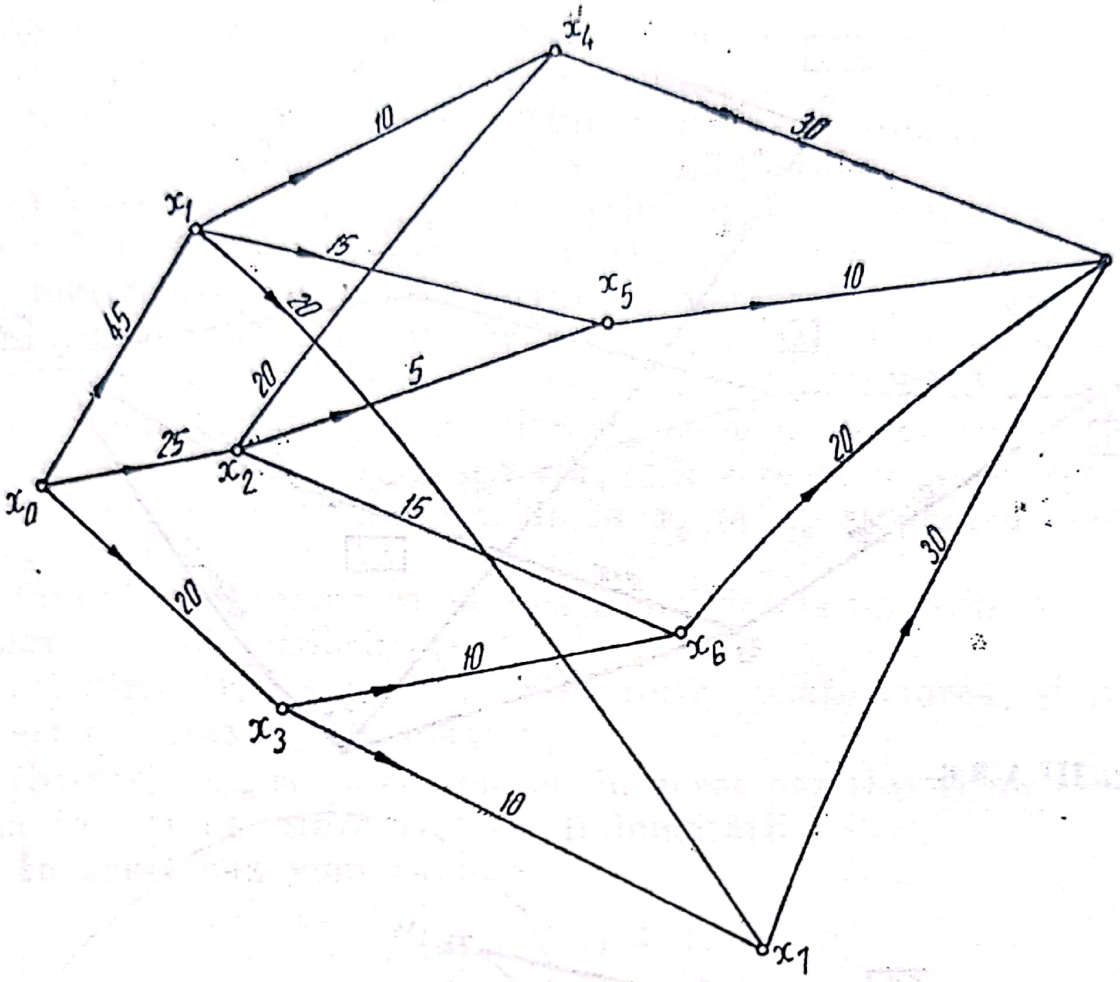


Fig. III.A.3.5

(x_0, x_1, x_5, x_2, x_4, x_8) cu încă 5 unităţi. Fluxul φ_1 îmbunătăţit este redat în figura III.A.3.7. şi este

$$\varphi_1 = 80 + 5 = 85.$$

Trecînd din nou la marcarea vîrfurilor grafului, constatăm că nu mai poate fi marcată ieşirea din reţea x_8, şi ca atare, fluxul φ_1 este fluxul maxim.

Analizînd soluţia problemei, constatăm că ea este unică şi că pentru satisfacerea debitului necesar de 30 l/s în C.A.P. x_4, folosind disponibilul de 5 l/s din bazinul x_1 (rămas neutilizat), este necesar să se construiască un canal suplimentar între bazinul x_1 şi C.A.P. x_4, cu un debit de cel puţin 5 l/s.

2°. Există şi o altă variantă a algoritmului Ford-Fullkerson, care se deosebeşte de cea prezentată fie prin modul

Scanned with OKEN Scanner

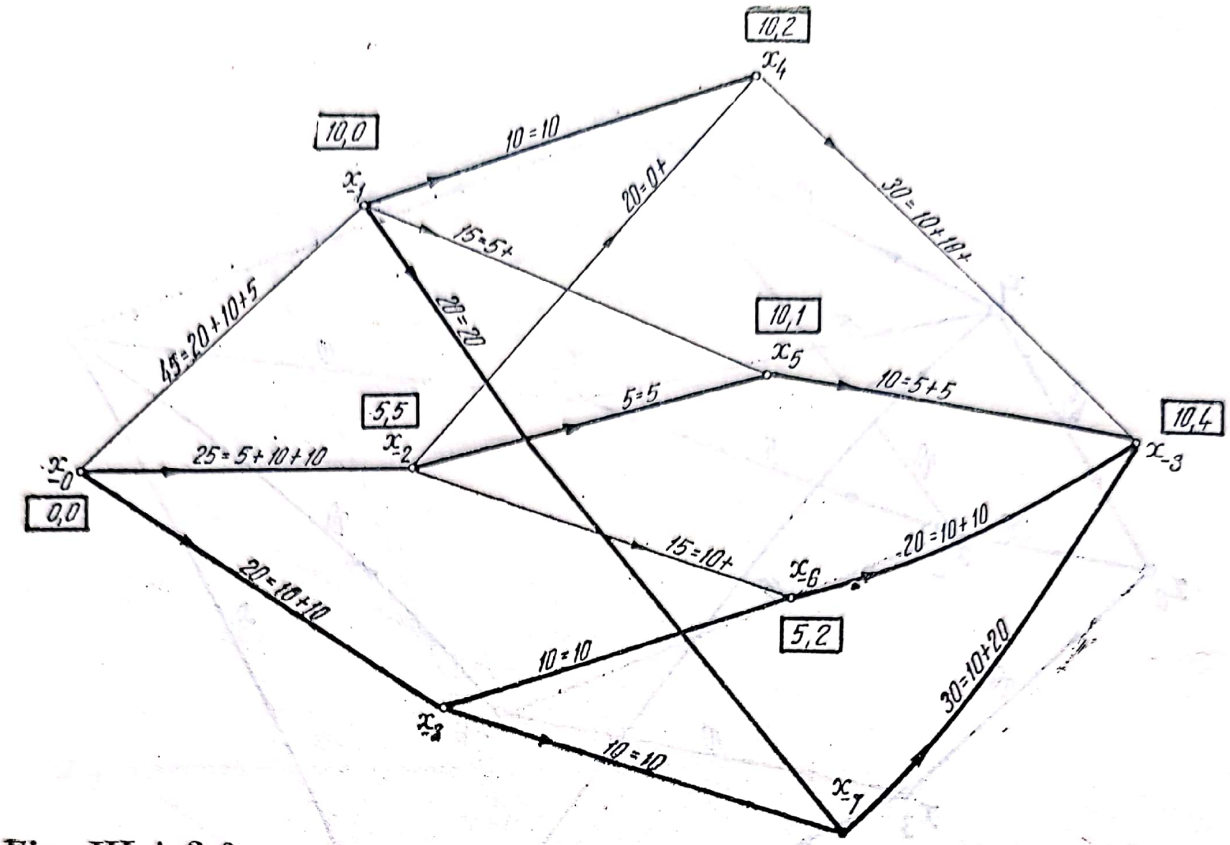


Fig. III.A.3.6

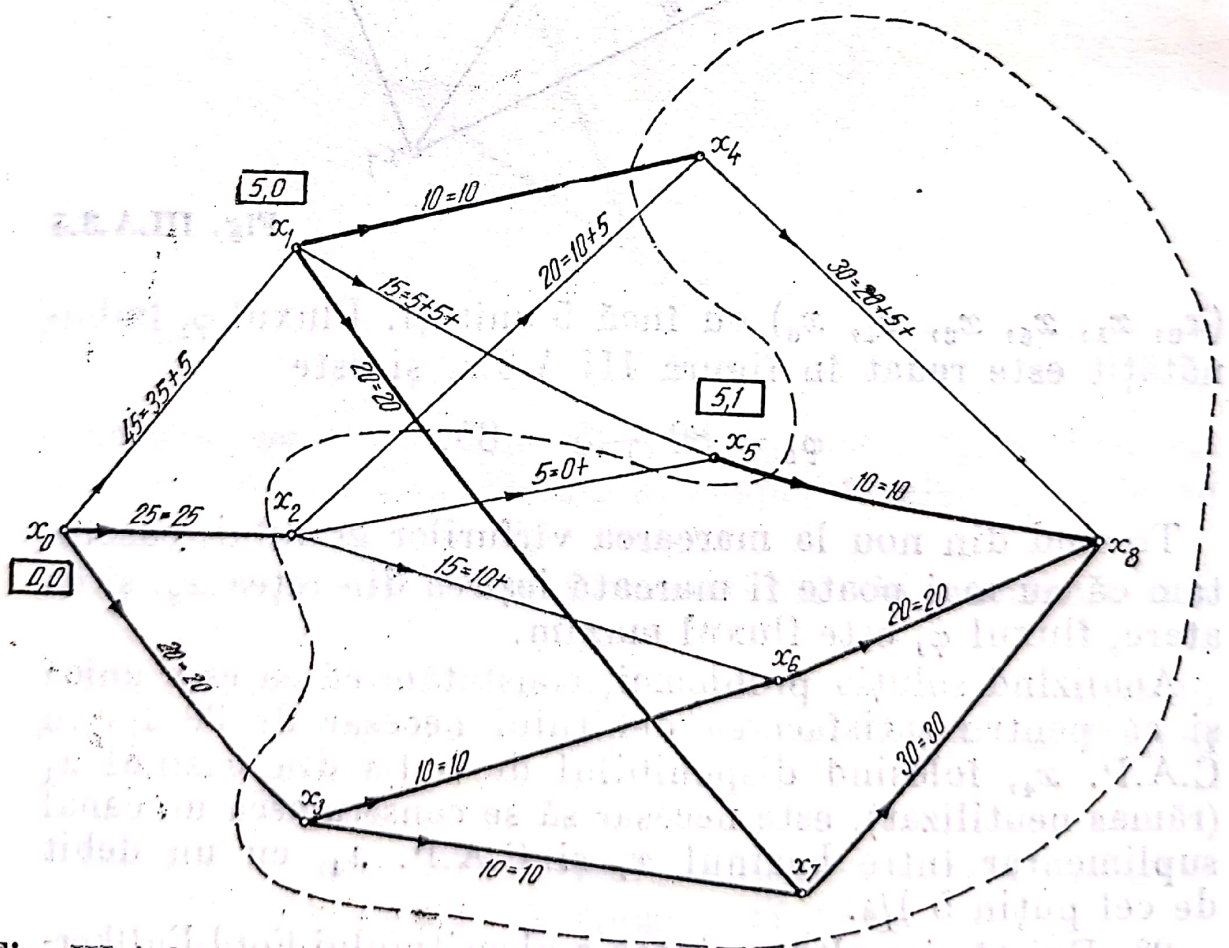


Fig. III.A.3.7

diferit în care se face marcajul, fie înlocuind întru totul procedeul de marcare prin alte operații convenabile.

Un alt mod de a marca vîrfurile grafului după ce a fost găsit un flux complet constă în următoarele:

a) Marcăm vîrful x_0 (intrarea în rețea) cu semnul (+), apoi din aproape în aproape dacă x_i este un vîrf marcat, se marchează cu $(+i)$ vîrful x_j nemarcat, dacă arcul $(x_i, x_j) = a_{ij} \in A$ și este orientat în sensul de parcurs al drumului de la x_0 la x_n și pentru care $f(a_{ij}) < C(a_{ij})$.

b) Dacă vîrful x_j este marcat, se marchează cu $(-j)$ vîrful x_k, cînd arcul $x_k, x_j = a_{kj} \in A$ este orientat în sens invers sensului de parcurs de la x_0 la x_n și pentru care $f(a_{kj}) > 0$.

Procedeul de marcare ne poate conduce la una din următoarele două posibilități:

(a) vîrful x_n ieșirea din rețea nu se poate marca, și în acest caz fluxul este maxim;

(b) vîrful x_n se poate marca în acest caz fluxul nu este maxim, și ca atare trebuie îmbunătățit.

În acest caz vom calcula

$$f'(a) = f(a) + 1,$$

dacă arcul a aparține drumului marcat de la x_0 la x_n în sensul de parcurs și

$$f'(a) = f(a) - 1,$$

dacă arcul a aparține drumului marcat, dar este orientat în sens invers sensului de parcurs.

Pentru celelalte arce care nu aparțin drumului marcat de la x_0 la x_n luăm

$$f'(a) = f(a).$$

Exemplu: Fie graful din figura III.A.3.8.

Să presupunem că fluxul inițial construit este indicat pe același graf cu culoare roșie.

Procedînd la marcarea vîrfurilor grafului, constatăm că se poate marca vîrful x_{11}; în această situație fluxul nu este maxim.

Într-adevăr, pe traseul $(x_0, x_3, x_2, x_1, x_4, x_5, x_8, x_{11})$ avem posibilitatea de a crește cu o unitate în condițiile arătate,

Scanned with OKEN Scanner

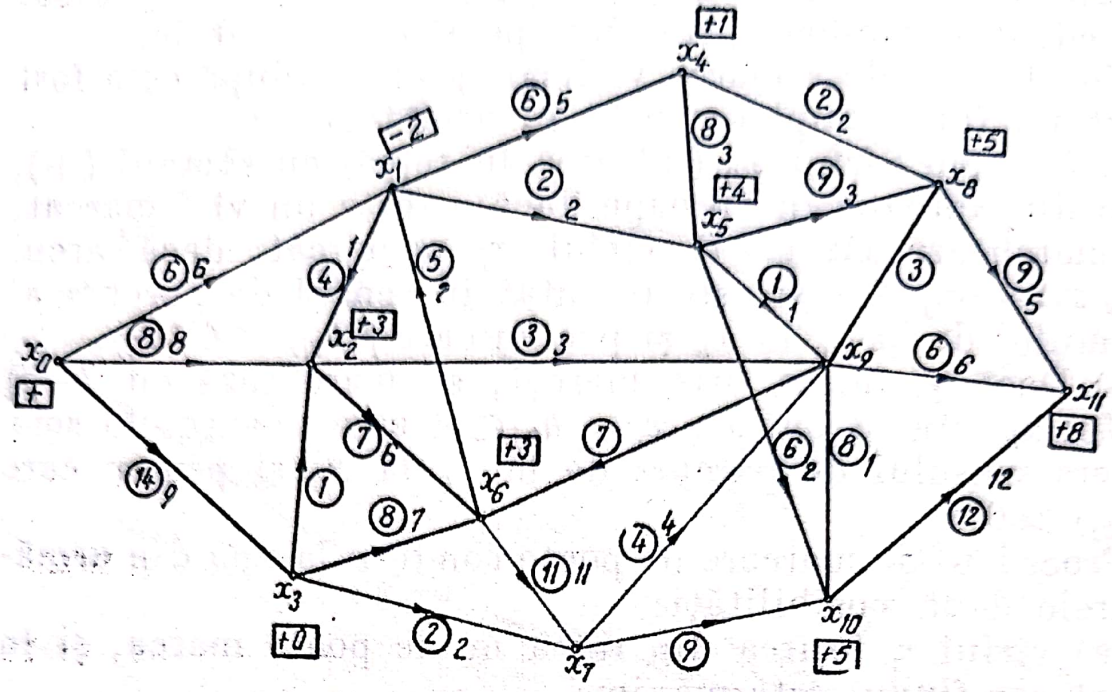


Fig. III.A.3.8

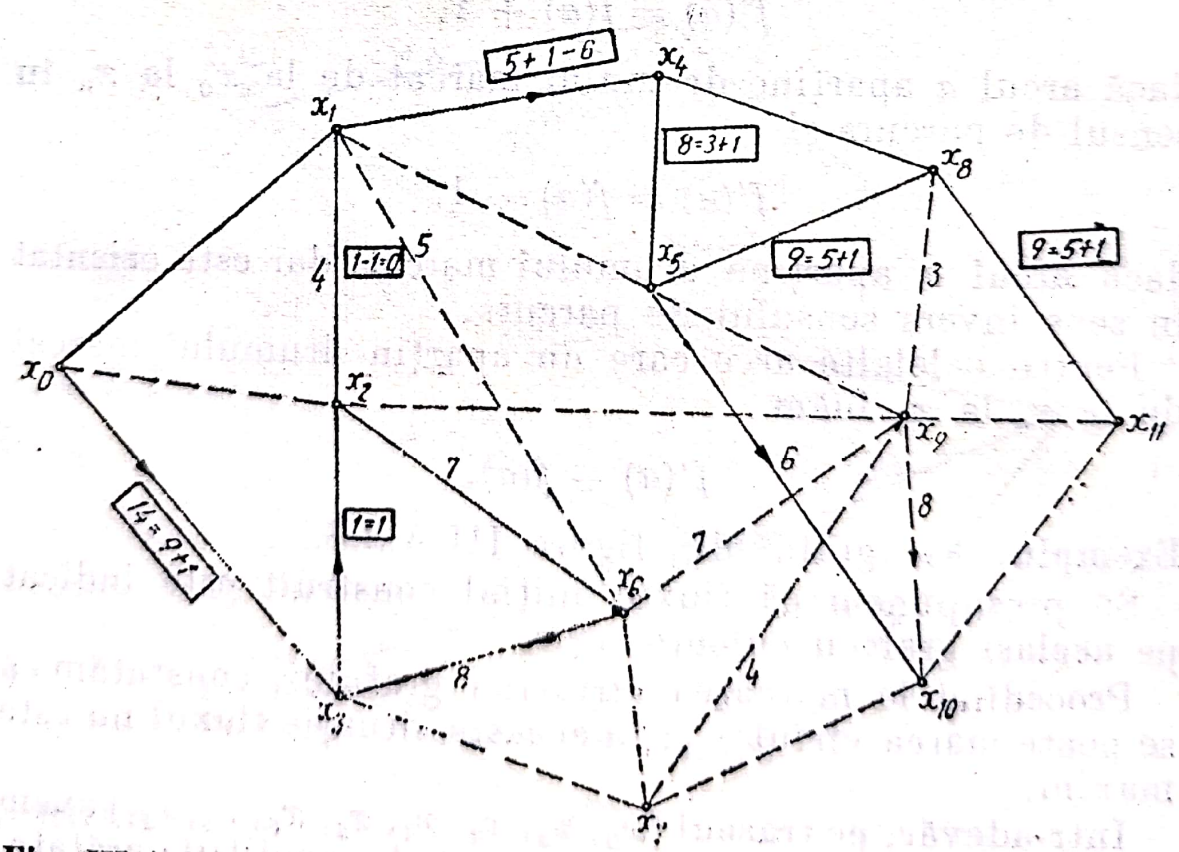


Fig. III.A.3.9

ceea ce va duce în final la mărirea fluxului (x_{11}) cu o unitate și la saturarea arcelor (x_3, x_2), (x_1, x_4).

Reluînd operația de marcare a vîrfurilor gradului redat în figura III.A.3.9 constatăm că vîrful x_{11} nu mai poate fi marcat, de unde deducem că s-a obținut fluxul maxim,

$$f_{max} = 24.$$

Algoritmul pe care l-am prezentat are la bază teorema Ford-Fullkerson, care afirmă că „fluxul maxim într-o rețea de transport este egal cu capacitatea minimă a unei tăieturi", adică:

$$\max f_n = \min C(A_y^-), \text{ cu } x_0 \notin Y,\ x_n \in Y.$$

DEMONSTRAȚIE:

Într-adevăr, fie $Y \subset X$, cu $x_0 \notin Y$ și $x_n \in Y$. Notăm complimentara în raport cu X, a submulțimii Y cu $\bar{Y}$. Din modul în care a fost ales Y rezultă că mulțimea $\bar{Y}$ conține intrarea în rețea x_0, și nu conține ieșirea. Cu alte cuvinte, au loc relațiile:

$$x_0 \in \bar{Y},\ x_n \notin \bar{Y},\ \text{iar } \bar{Y} = X - Y = C_X Y.$$

Plecînd de la relația cunoscută

$$f_n = \sum_{a \in A_{x_0}^+} f(a) = \sum_{a \in A_{x_n}^-} f(a), \tag{1}$$

în care prima sumă o putem scrie și sub forma

$$f_n = \sum_{a \in A_{x_0}^+} f(a) = \sum_{j=1}^{n} f(a_{0_j}), \tag{2}$$

deoarece am notat cu a_{0_j} arcele incidente spre exterior vîrfului x_0.

Am arătat că într-o rețea nu există pierderi, adică pentru

orice vîrf $\begin{cases} x_i \in X \\ x_i \neq x_0 \\ x_i \neq x_n \end{cases}$ există egalitatea

$$\sum_{a \in A_{x_i}^+} f(a) = \sum_{a \in A_{x_i}^-} f(a). \tag{3}$$

Scanned with OKEN Scanner

Relația (3), dacă se ține seama de toate vîrfurile submulțimii $\bar{Y}$, mai puțin vîrful x_0, mai poate fi scrisă încă:

$$\sum_{j=1}^{n} \sum_{i \in \bar{Y} - \{x_0\}} f(a_{ij}) = \sum_{j=1}^{n} \sum_{i \in \bar{Y} - \{x_0\}} f(a_{ji}) \tag{4}$$

Altfel spus, suma fluxurilor care pleacă din mulțimea $\bar{Y} - \{x_0\}$ este egală cu suma fluxurilor care vine în aceeași mulțime.

În aceste condiții, relația (2) devine:

$$f_n = \sum_{j=1}^{n} f(a_{0_j}) + \sum_{j=1}^{n} \sum_{i \in \bar{Y} - \{x_0\}} f(a_{ij}) - \sum_{j=1}^{n} \sum_{i \in \bar{Y} - \{x_0\}} f(a_{ji}) \tag{5}$$

deoarece diferența ultimilor doi termeni din membrul al doilea al egalității (5) este nulă.

Relația (5) se scrie încă:

$$f_n = \sum_{j=1}^{n} [f(a_{0_j}) + \sum_{i \in \bar{Y} - \{x_0\}} f(a_{ij}) - \sum_{i \in \bar{Y} - \{x_0\}} f(a_{ji}) =$$

$$= \sum_{j=1}^{n} \sum_{i \in \bar{Y}} [f(a_{ij}) - f(a_{ji})]. \tag{6}$$

Descompunem acum suma referitoare la indicele j în două sume și anume: una referitoare la indicii j corespunzători vîrfurilor $x_j \in Y$ și una referitoare la indicii j corespunzători vîrfurilor $x_i \in \bar{Y}$.

În aceste condiții, relația (6) se poate scrie:

$$f_n = \sum_{i \in \bar{Y}} \sum_{j=1}^{n} [f(a_{ij}) - f(a_{ji})] =$$

$$= \sum_{i \in \bar{Y}} \sum_{j \in \bar{Y}} [f(a_{ij}) - f(a_{ji})] + \sum_{j \in \bar{Y}} [f(a_{ij}) - f(a_{ji})] =$$

$$= \sum_{i \in \bar{Y}} \sum_{j \in \bar{Y}} [f(a_{ij}) - f(a_{ji})] + \sum_{i \in \bar{Y}} \sum_{j \in \bar{Y}} [f(a_{ij}) - f(a_{ji})]; \tag{7}$$

dar

$$\sum_{i \in \bar{Y}} \sum_{j \in \bar{Y}} f(a_{ij}) - f(a_{ji}) =$$

$$= \sum_{i \in \bar{Y}} \sum_{j \in \bar{Y}} f(a_{ij}) - \sum_{i \in \bar{Y}} \sum_{j \in \bar{Y}} f(a_{ji}) = 0,$$

deoarece fluxul care intră în Y este egal cu fluxul care iese din Y.

Atunci relația (7) se scrie:

$$f_n = \sum_{i \in \bar{Y}} \sum_{j \in \bar{Y}} [f(a_{ij}) - f(a_{ji})].$$

Știm însă că fluxul care parcurge arcul a_{ij} este mai mic, cel mult egal cu capacitatea aceluiași arc.

Prin urmare

$$f_n = \sum_{i \in Y} \sum_{j \in Y} [f(a_{ij}) - f(a_{ji})] \leqslant \sum_{i \in \bar{Y}} \sum_{j \in \bar{Y}} f(a_{ij}) \leqslant$$

$$\leqslant \sum_{i \in \bar{Y}} \sum_{j \in \bar{Y}} c(a_{ij}) \qquad (8)$$

Ultimul termen al inegalității (8) reprezintă suma capacităților arcelor care au extremitatea inițială în mulțimea $\bar{Y}$ și extremitatea finală în mulțimea Y, adică tocmai capacitatea tăieturii A_Y^- și deci:

$$f_n \leqslant C(A_Y^-). \qquad (9)$$

Fie acum f_n^* fluxul maxim în rețeaua de transport.

Considerăm toate drumurile posibile care pleacă din x_0, folosite în determinarea fluxului maxim, f_n^*, și pentru care arcele din componența lor satisfac relația

$$f^*(a_{ij}) - f^*(a_{ji}) < C(a_{ij}). \qquad (10)$$

Altfel spus, luăm în considerație drumurile care pleacă din x_0, care conțin arce nesaturate. Evident că orice drum de acest gen nu poate avea extremitatea finală în x_n, deoarece în acest caz fluxul f_n^* ar putea fi îmbunătățit, ceea ce contrazice presupunerea făcută că f_n^* este fluxul maxim în rețea.

Mulțimea $\bar{Y}$ a extremităților arcelor pentru care relația (10) este satisfăcută conține intrarea în rețea, dar nu conține ieșirea.

Fie Y complementara în X a mulțimii $\bar{Y}$. Mulțimea arcelor A_Y^- am arătat că formează o tăietură, pentru care avem satisfăcută relația

$$f_n^* < C(A_Y^-) \qquad (11)$$

Scanned with OKEN Scanner

sau $\sum_{i\in\bar{Y}}\sum_{j\in\bar{Y}}[f^*(a_{ij}) - f^*(a_{ji})] < \sum_{i\in\bar{Y}}\sum_{j\in\bar{Y}} c(a_{ij})$. (12)

Din relaţia (12) rezultă existenţa a cel puţin două vîrfuri $x_p \in \bar{Y}$, şi $x_q \in Y$ pentru care

$$f^*(a_{pq}) - f^*(a_{qp}) < C(a_{pq}).$$

Din felul cum a fost definită mulţimea $\bar{Y}$ rezultă că ambele extremităţi ale arcului a_{pq} aparţin mulţimii Y, ceea ce contrazice ipoteza făcută ca $x_p \in \bar{Y}$ şi $x_q \in \bar{Y}$.

Contrazicerea apărută rezultă din faptul că am presupus relaţia (11) adevărată.

Din această contradicţie şi din faptul că fluxul nu depăşeşte capacitatea tăieturii, nu rămîne decît să admitem că:

$$f_n^* = C(A_{\bar{Y}}).$$

Mai rămîne de arătat că tăietura $A_{\bar{Y}}$, astfel definită, are capacitatea minimă.

Admitem prin absurd că ar exista o altă tăietură $A_{\bar{Y}'}$ a cărei capacitate este mai mică, adică

$$C(A_{\bar{Y}'}) < C(A_{\bar{Y}}). \quad (13)$$

Ştim însă că orice flux care străbate reţeaua este mai mic, cel mult egal cu capacitatea oricărei tăieturi, şi deci:

$$f_n^* \leqslant C(A_{\bar{Y}'}) < C(A_{\bar{Y}}). \quad (14)$$

Din relaţia (14) rezultă:

$$f_n^* < C(A_{\bar{Y}}). \quad (15)$$

Relaţia (15) nu poate avea loc, datorită existenţei relaţiei

$$f_n^* = C(A_{\bar{Y}}).$$

demonstrată mai sus.

În concluzie, $A_{\bar{Y}}$ este tăietura de capacitate minimă.

Cu aceasta, teorema Ford-Fullkerson este demonstrată.

Verificăm teorema Ford-Fullkerson considerînd graful din figura III.A.3.2′, în care este redat fluxul maxim în reţeaua corespunzătoare exemplului (1). Mulţimea X a vîrfurilor grafului a fost împărţită cu ajutorul procedeului de marcare

Scanned with OKEN Scanner

în două submulţimi, unde într-una din submulţimi găsim vîrfurile marcate, printre care şi intrarea în reţea, iar în cealaltă, vîrfurile nemarcate, care conţine evident şi ieşirea din reţea. Notînd cu Y mulţimea vîrfurilor nemarcate

$$Y = \{x_1, x_2, x_4, x_5, x_6, x_7\},$$

tăietura corespunzătoare este

$$A_Y^- = \{(x_0, x_1), (x_0, x_2), (x_3, x_5), (x_3, x_6)\},$$

iar capacitatea acestei tăieturi,

$$C(A_Y^- (= c(x_0, x_1) + c(x_0, x_2) + c(x_3, x_5) + (x_3, x_6) = \\ = 8 + 5 + 2 + 4 = 19,$$

egală cu fluxul maxim stabilit.

Analog, în graful din figura II.A.3.6, care redă fluxul maxim din reţeaua corespunzătoare exemplului (2), avem:

$$Y = \{x_2, x_3, x_4, x_6, x_7, x_8\}.$$

$$A_Y^- = \{(x_1, x_4); (x_1, x_7); (x_5, x_8); (x_0, x_2); (x_0, x_3)\}.$$

$$C(A_Y^-) = c(x_1, x_4) + c(x_1, x_7) + c(x_5, x_8) + c(x_0, x_2) + \\ + c(x_0, x_3) = 10 + 20 + 10 + 25 + 20 = 85 = f_{\max}.$$

3°. Cele două variante prezentate impun într-o formă sau alta operaţia de marcare a vîrfurilor grafului pentru care mai întîi s-a determinat un flux complet.

Prezentăm în continuare un algoritm care evită procedeul de marcare. Pentru aceasta, considerăm reţeaua de transport $G = (X, F)$.

Luînd ca punct de plecare teorema Ford-Fullkerson, care afirmă că fluxul maxim în reţea este egal cu capacitatea minimă a tăieturii, determinăm printr-un procedeu iterativ graful $G_k = (X, F)$, în care să existe o tăietură Y de capacitate nulă, $C(A_Y^-) = 0$.

În esenţă, algoritmul propus constă în a construi grafe de forma:

$$G^{(k)} = (X, F)$$

în care capacităţile arcelor sînt date de relaţia

$$C_{(a)}^{(k)} = C_{(a)}^{(k-1)} - f_{(a)}^{(k-1)};\ k = 1,\ 2,\ \ldots,\ p.$$

Dacă după un număr oarecare de paşi am găsit graful $G^{(p)} = (X,\ F)$, în care oricare ar fi drumul de la x_0 la x_n, el conţine cel puţin un arc de capacitate nulă, operaţia de construire a grafelor G_k se termină.

În graful $G^{(p)}$ nemaiputînd fi transportată nici o unitate de la x_0 la x_n, deoarece există pe fiecare drum un arc de capacitate nulă, am obţinut o tăietură de capacitate nulă şi evident minimă.

Tăietura de capacitate nulă din graful $G^{(p)}$ corespunde de fapt tăieturii de capacitate minimă din graful iniţial.

Revenind în graful $G^{(p)}$ la capacităţile iniţiale, capacitatea acestei tăieturi ne va da fluxul maxim în reţeaua $G = (X, F)$.

$$f_{\max} = f^{(0)} + f^{(1)} + \ldots + f^{(p)}.$$

Ilustrăm procedeul, determinînd fluxul maxim în reţeaua din figura III.A.3.10.

Construim un flux iniţial $f^{(0)}$, urmărind saturarea a cît mai multe arce (figura III.A.3.11).

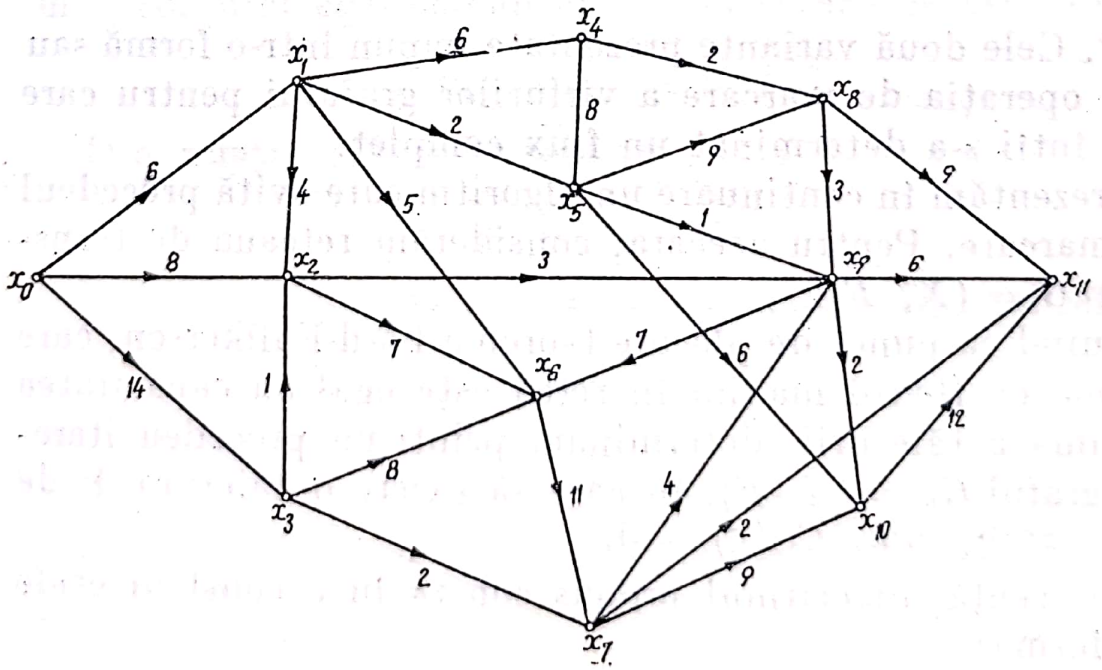


Fig. III.A.3.10

Scanned with OKEN Scanner

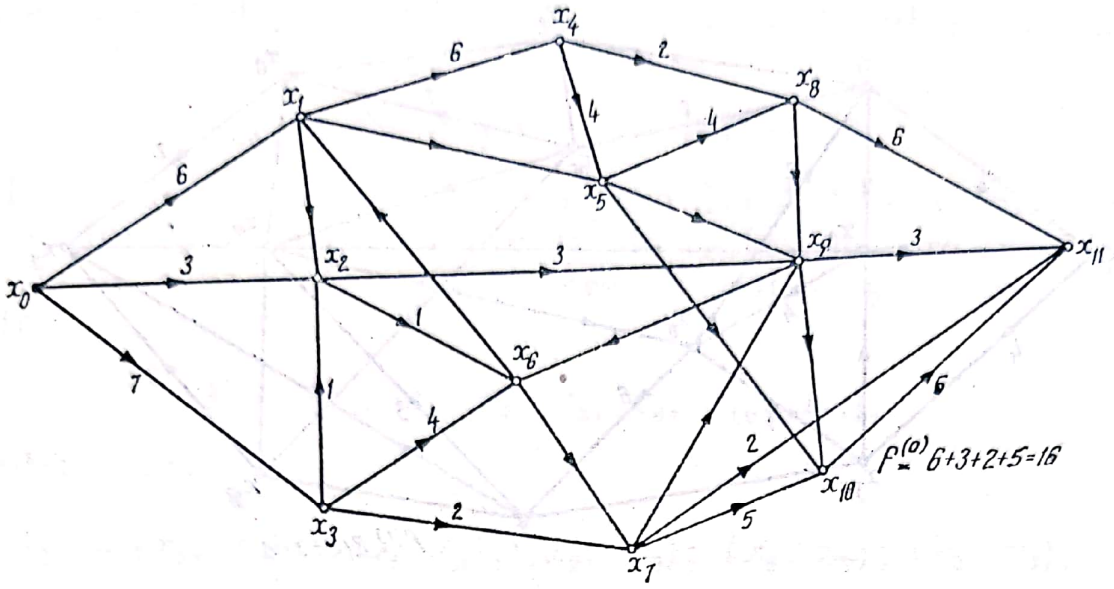


Fig. III.A.3.11

Construim graful G_1 cu capacitățile (figura III.A.3.12):

$$C^{(1)}_{(a)} = C^{(0)}(a) - f^{(0)}_{(a)}.$$

În graful $G^{(1)}$ construim un flux, urmărind de asemenea saturarea a cît mai multor arce. Obținem în acest fel graful din figura III.A.3.13.

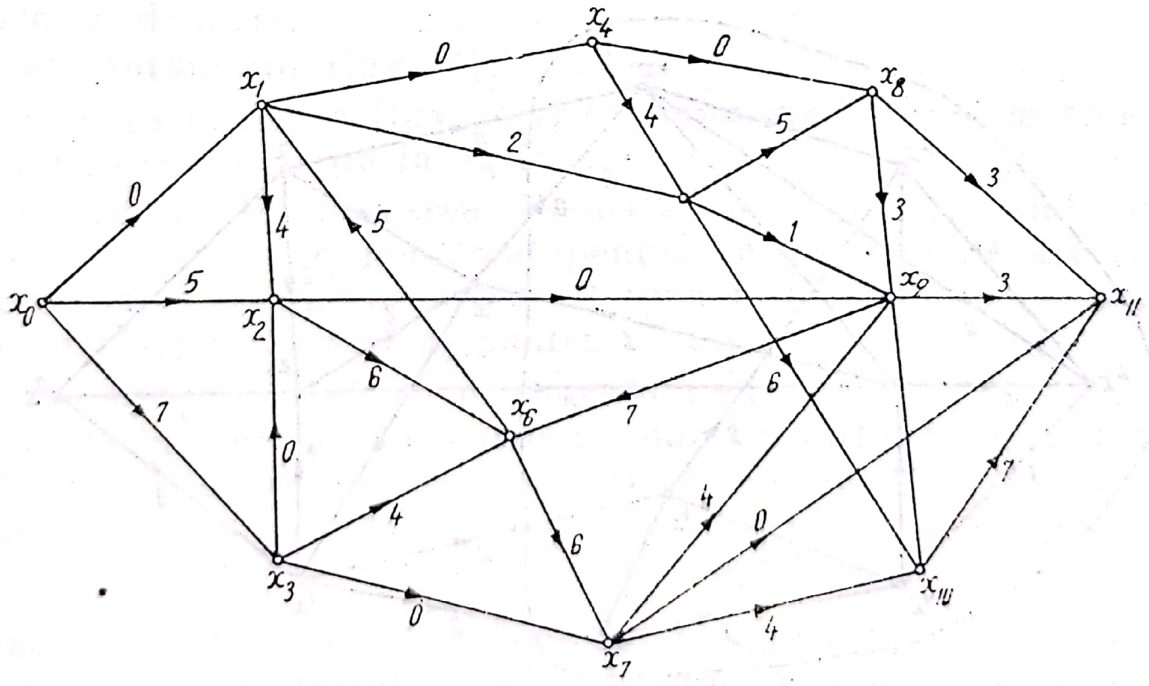


Fig. III.A.3.12

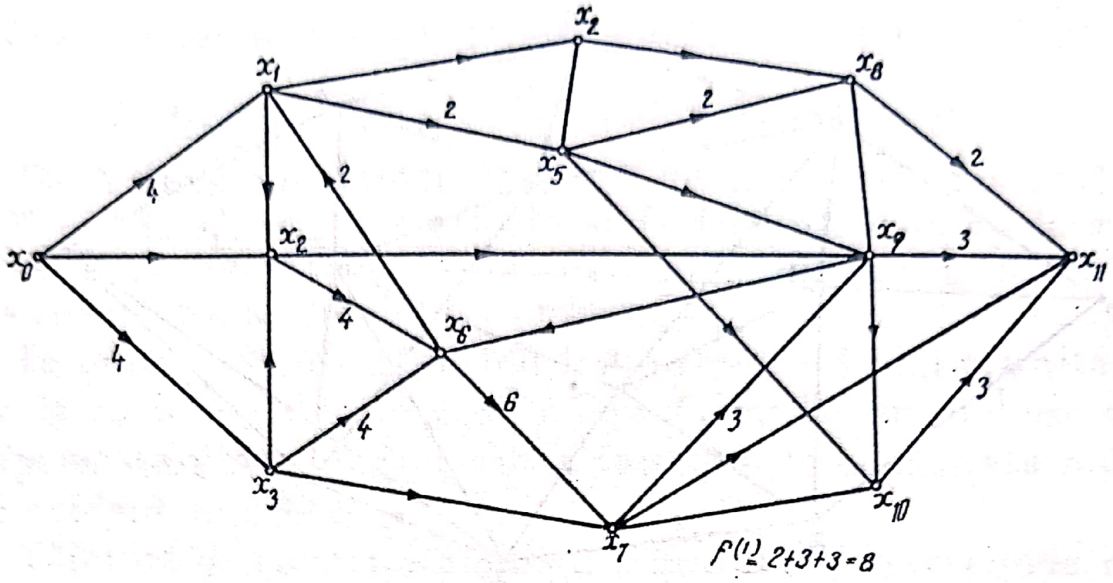


Fig. III.A.3.13

Construim din nou, graful $G^{(2)} = (X, F)$ avînd capacităţile $C^{(2)}(a) = C^{(1)}(a) - f^{(1)}(a)$ redat în figura III.A.3.14.

În graful $G^{(2)}$ constatăm că oricare ar fi drumul de la x_0 la x_{11}, el conţine cel puţin un arc de capacitate nulă. Pe oricare din aceste drumuri, avînd arce de capacitate nulă nu mai este posibilă transportarea nici unei unităţi;

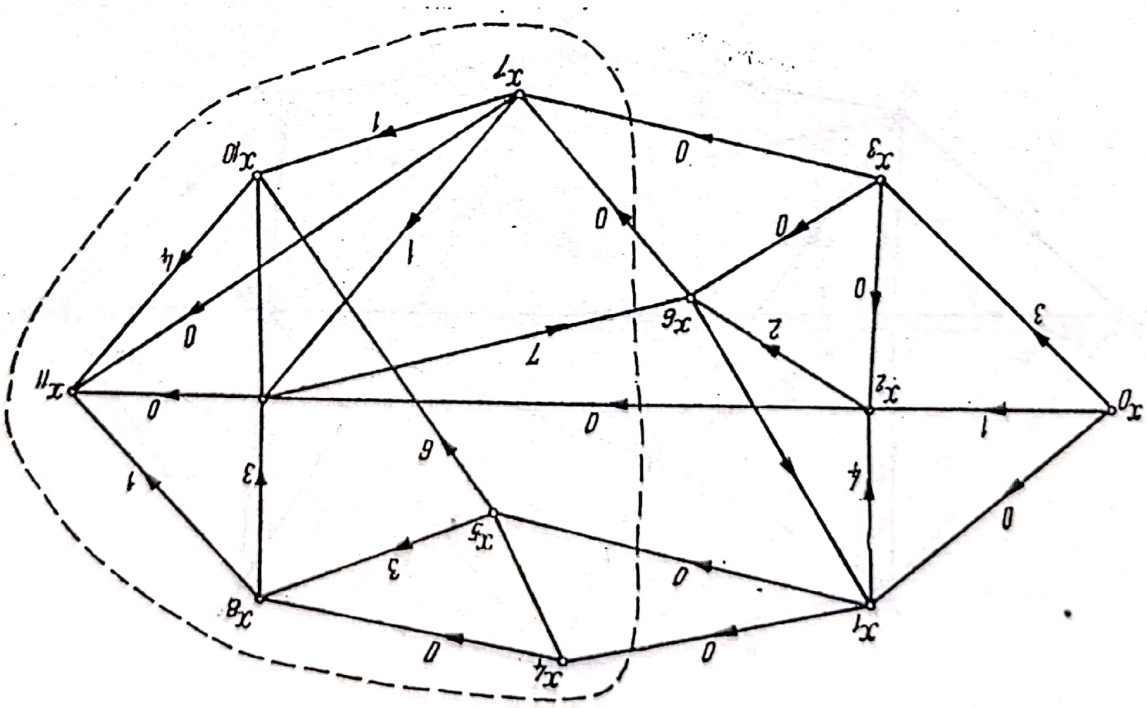


Fig. III.A.3.14

în aceste condiţii, rezultă că s-a obţinut fluxul maxim în reţea şi

$$f_{\max} = f_1^{(0)} + f^{(1)} = 16 + 8 = 24.$$

Luăm acum în graful din figura III.A.3.14 o mulţime de vîrfuri pentru care toate arcele incidente spre interior să fie de capacitate nulă şi care să conţină şi ieşirea din reţea, adică

$$Y = \{x_4, x_5, x_7, x_8, x_9, x_{10}, x_{11}\},$$

pentru care

$$A_Y^- = (x_1, x_4); (x_1, x_5); (x_2, x_9); (x_6, x_7); (x_3, x_7),$$

unde:

$$C_0(A_Y^-) = C(x_1, x_4) + C(x_1, x_5) + C(x_2, x_9) + C(x_6, x_7) + C(x_3, x_7) = 6 + 2 + 3 + 11 + 2 = 24.$$

De unde rezultă că fluxul găsit este maxim.

4°. Din cele prezentate pînă acum nu trebuie să rezulte că practica cere numai căutarea fluxului maxim, şi uneori este nevoie să se determine şi fluxul minim. În asemenea situaţii, algoritmului Ford-Fullkerson îi aducem următcarele modificări:

a) căutăm un flux $f(a) \geqslant C(a)$,

b) construim un flux complet, micşorînd fluxurile arcelor pe drumurile de la x_0 la x_n,

c) trecem la marcarea vîrfurilor grafului, marcînd cu $(+i)$ orice vîrf x_j nemarcat pentru care există un arc în care fluxul este mai mare decît capacitatea arcului (x_i, x_j) şi cu $(-i)$ orice vîrf x_j pentru care există arcul (x_j, x_i).

Intrarea în reţea se marchează $(+)$.

Dacă în acest fel nu reuşim să marcăm vîrful (x_n), fluxul obţinut este minim. În caz contrar, continuăm cu micşorarea fluxului pe traseele marcate pînă cînd nu mai poate fi marcat vîrful x_n (ieşirea din reţea).

Observaţie: Evident că în cazul determinării fluxului minim capacităţile reprezintă limita inferioară a numărului de unităţi ce străbate arcul.

Scanned with OKEN Scanner

§ 4. PROBLEME CARE CONDUC LA DETERMINAREA FLUXULUI MAXIM ÎN REȚEA

1°. Presupunem că în centrele Curtea de Argeș, Făgăraș, Măneciu Ungureni și Bistrița se găsesc la un moment dat următoarele cantități de cherestea:

240 t; 200 t; 200 t și 200 t.

Această cantitate urmează să fie transportată la patru centre de consum $C_{i(i=1, 2, 3, 4)}$ în cantitățile respective:

200 t; 160 t; 180 t; 300 t.

Posibilitățile de transport, limitate de capacitățile mijloacelor de transport sînt date de tabelul de mai jos:

	Notații / Centre de destinație / Centre de expediție	x_1^* / C_1	x_2^* / C_2	x_3^* / C_3	x_4^* / C_4
x_1	Curtea de Argeș	140	60	40	40
x_2	Făgăraș	100	80	20	0
x_3	Măneciu Ungureni	0	40	80	160
x_4	Bistrița	0	40	80	160

Numărul 140 din căsuța situată pe linia întîi și coloana întîi, arată că mijloacele de transport existente oferă posibilitatea de a transporta 140 t cherestea între Curtea de Argeș și C_1. Existența unui zero dintr-o anumită căsuță indică faptul că pe ruta respectivă nu se face nici un transport.

Se poate presupune că unele centre au mai urgentă nevoie decît altele.

Scanned with OKEN Scanner

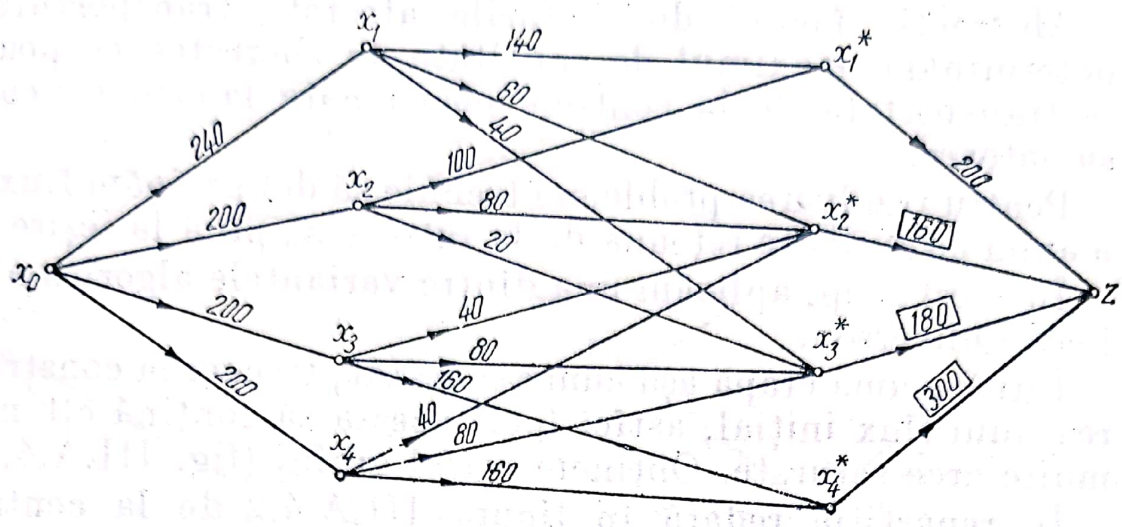


Fig. III.A.4.1

În exemplul luat vom presupune, de exemplu, că C_4 și C_2 trebuie satisfăcute în primul rînd.

Problema conduce la construirea unei rețele de transport de forma arătată în figura III.A.4.1.

Pe arcele care leagă centrul fictiv x_0 (intrarea în rețea) de centrele de expediere au fost trecute cantitățile disponibile acestor centre; iar pe arcele care leagă centrele de destinație cu punctul Z (ieșirea din rețea) au fost trecute cantitățile cerute de centrele consumatoare.

Evident că pe arcele care reprezintă ruta de transport au fost trecute capacitățile mijloacelor de transport existente.

Menționăm că valorile încercuite arată nevoia de a satisface în primul rînd centrele respective.

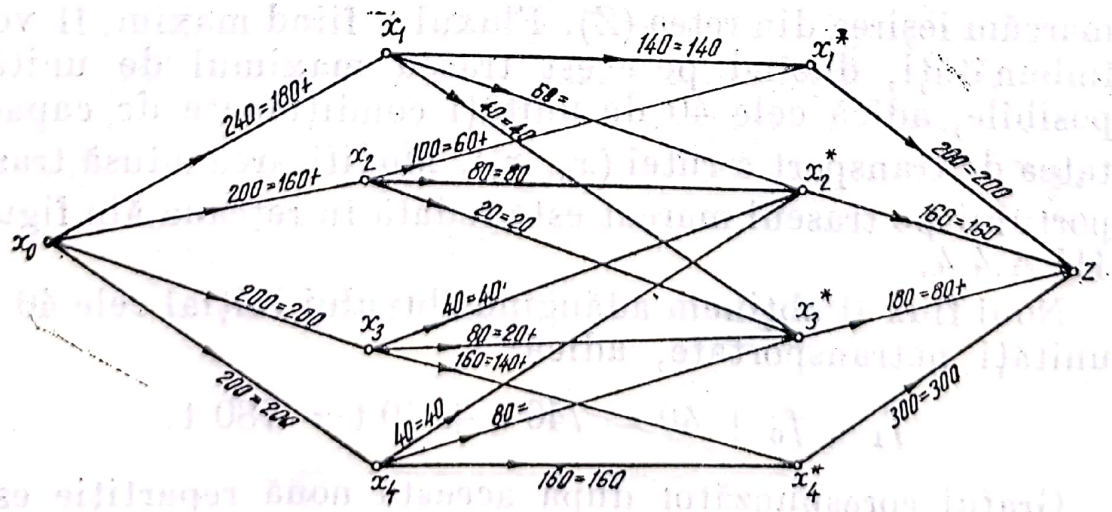


Fig. III.A.4.2

Abstracție făcînd de costurile afectate transportului, determinăm maximul de cantitate de cherestea ce poate fi transportată de la centrele menționate la centrele consumatoare.

Pentru rezolvarea problemei trebuie să determinăm fluxul maxim ce străbate rețeaua de la intrare x_0 pînă la ieșire Z.

În acest scop, aplicăm una dintre variantele algoritmului Ford-Fullkerson.

Într-o primă etapă așa cum s-a arătat, trecem la construirea unui flux inițial, astfel încît acesta să conțină cît mai multe arce saturate. Obținem astfel graful (fig. III.A.4.2).

În repartiția redată în figura III.A.4.2 de la centrul x_3 și x_4 a fost transportat tot disponibilul, centrele x_1^*, x_2^* și x_4^* au fost satisfăcute în întregime, iar pe rutele (x_1, x_1^*), (x_1, x_3^*), (x_2, x_2^*), (x_2, x_3^*), (x_3, x_2^*) și (x_4, x_4^*) au fost folosite toate mijloacele de transport disponibile și, ca atare, pe ele nu se mai poate transporta nici o unitate.

Evident că am avut grijă ca în primul rînd să satisfacem centrele cu prioritate x_2^* și x_4^*. Fluxul total în Z obținut după această primă repartiție este

$$f_0 = 200 + 160 + 80 + 300 = 740 \text{ t}.$$

Verificăm acum, prin procedeul de marcare a vîrfurilor, dacă această soluție este sau nu optimă, adică dacă s-a transportat maximul de cantitate posibilă.

Mergînd pe traseul $(x_0, x_1, x_2^*, x_3, x_3^*, Z)$, am reușit să marcăm ieșirea din rețea (Z). Fluxul nefiind maxim, îl vom îmbunătăți, ducînd pe acest traseu maximul de unități posibile, adică cele 40 de unități condiționate de capacitatea de transport a rutei (x_3, x_2^*). Modificarea adusă transportului pe traseul marcat este redată în rețeaua ăin figura III.A.4.4.

Noul flux îl obținem adăugînd fluxului inițial cele 40 de unități netransportate, adică:

$$f_1 = f_0 + 40 = 740 \text{ t} + 40 \text{ t} = 780 \text{ t}.$$

Graful corespunzător după această nouă repartiție este redat în figura III.A.4.5.

Scanned with OKEN Scanner

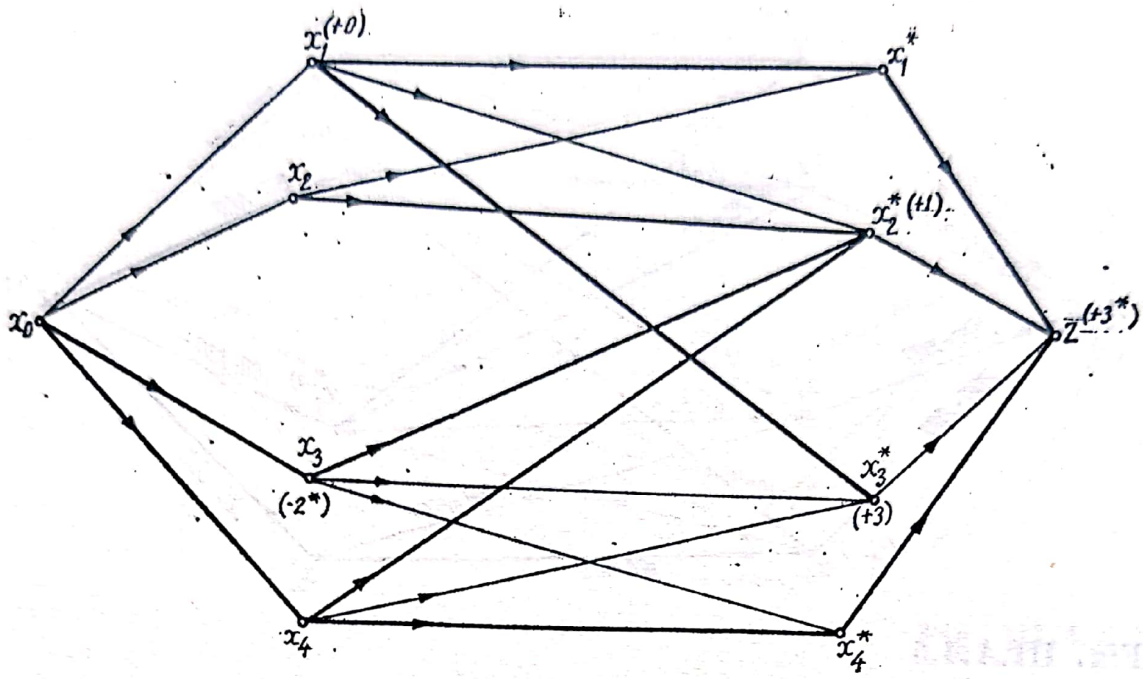


Fig. III.A.4.3

Pe graful din figura III.A.4.5, repetînd procedeul de marcare, am fost conduşi din nou la marcarea vîrfului Z şi ca atare fluxul este susceptibil de o nouă îmbunătăţire.

De această dată pe traseul $(x_0, x_1, x_2^*, x_4, x_3^*, Z)$ vom transporta numai 20 de unităţi (cantitate condiţionată de dispo-

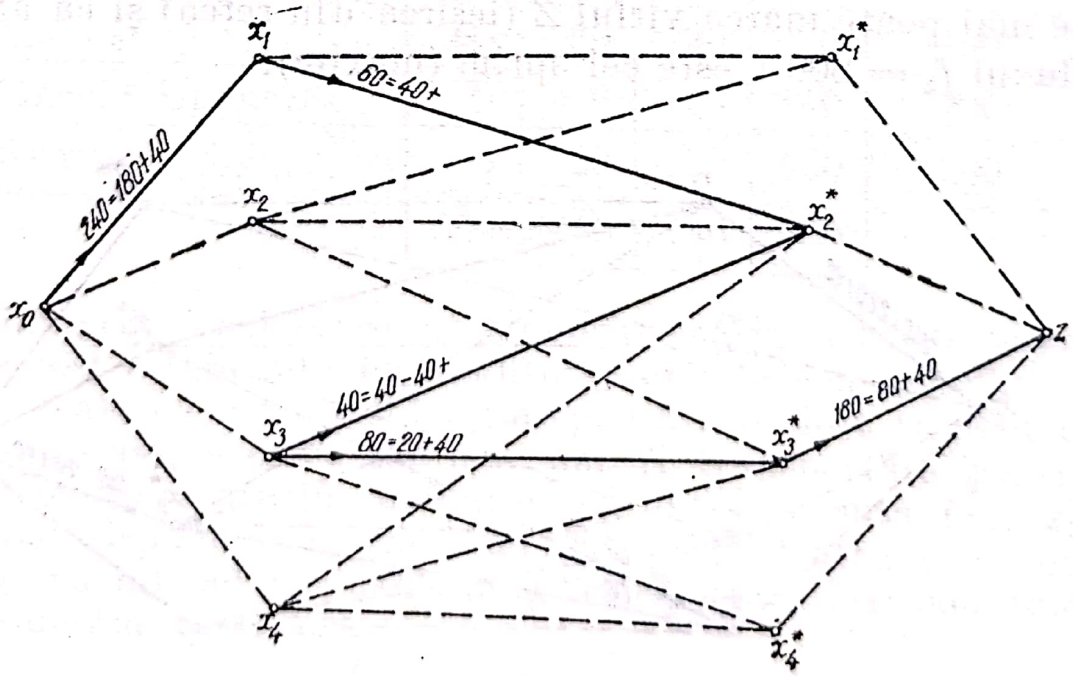


Fig. III.A.4.4

Scanned with OKEN Scanner

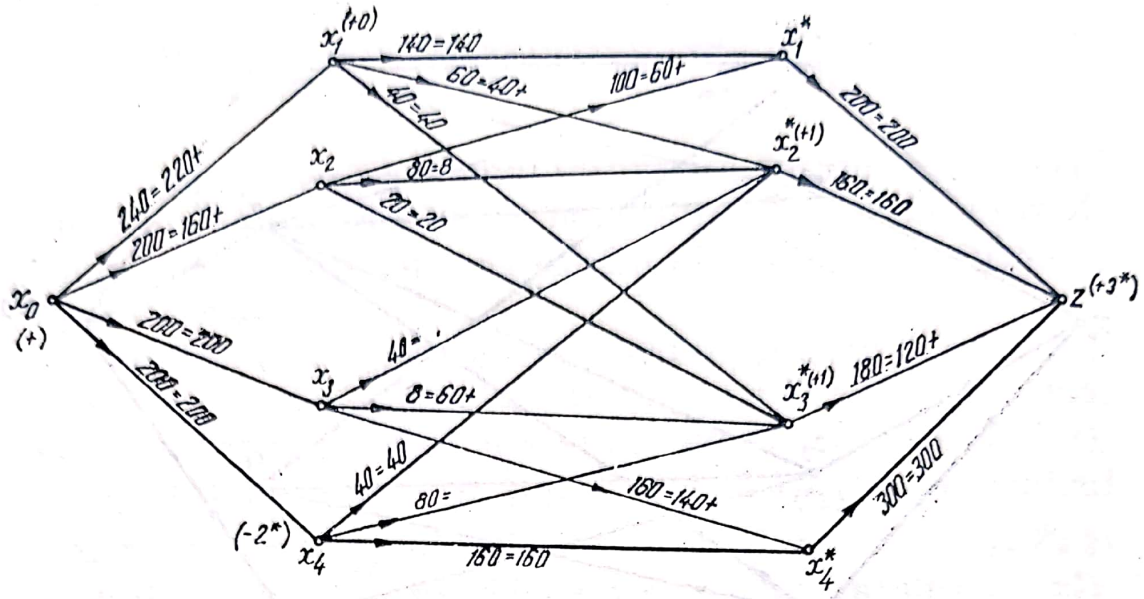


Fig. III.A.4.5

nibilul în centrul x_1, figura III.A.4.6 arătînd modificarea transportului pe traseul considerat).

Repartiția totală acum este înfățișată în rețeaua din figura III.A.4.7, iar

$$f_2 = f_1 + 20 = 780 + 20 = 800 \text{ t}.$$

Analizînd repartiția din ultimul graf, constatăm că nu se mai poate marca vîrful Z (ieșirea din rețea) și ca atare, fluxul $f_2 = 800$ t este cel optim (maxim).

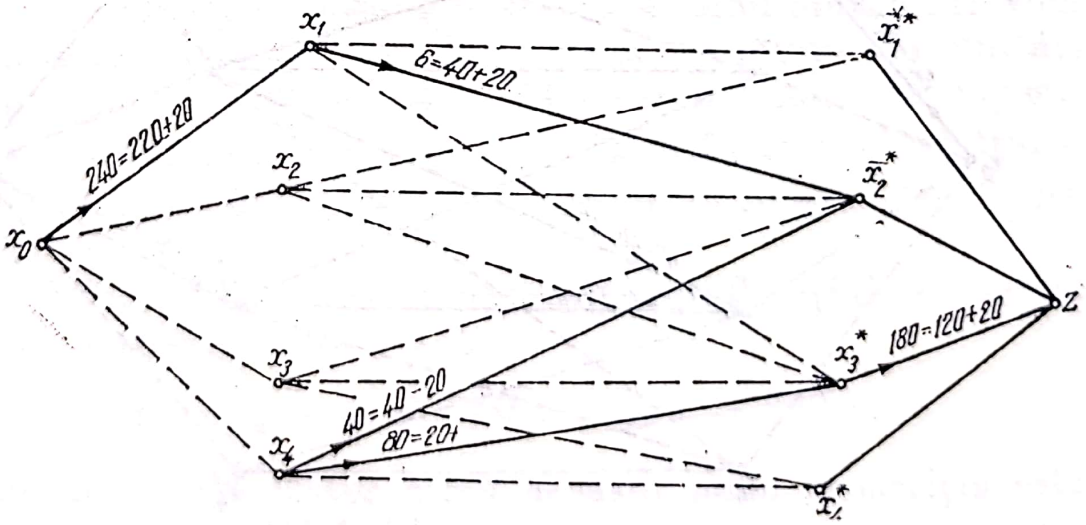


Fig. III.A.4.6

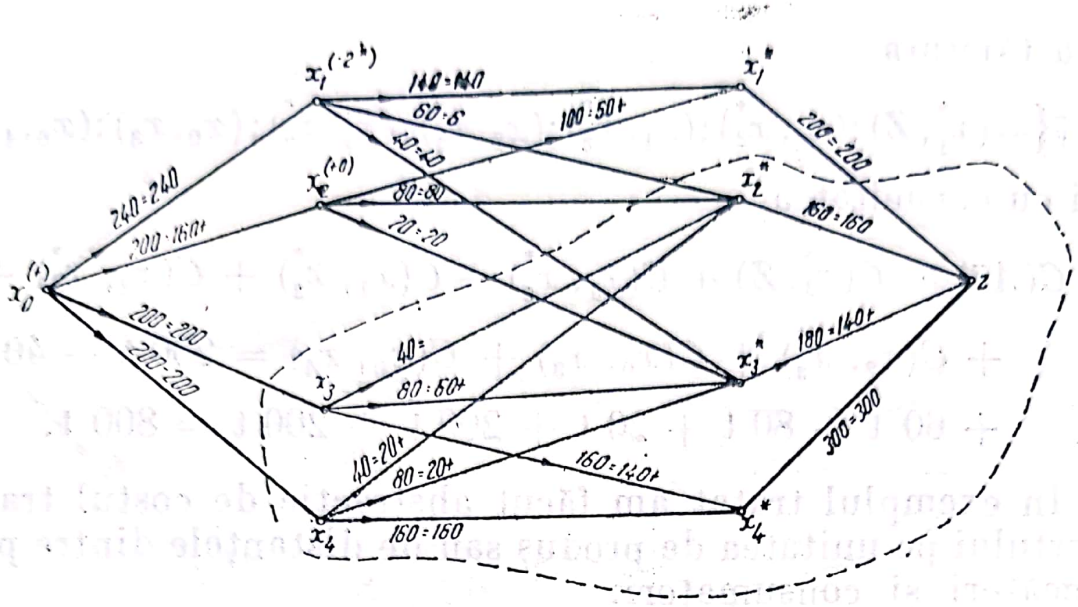


Fig. III.A.4.7

Repartiția optimă o redăm mai jos:

Centre de destinație / Centre de expediere	C_1	C_2	C_3	C_4	Cantități transportate din localități din țară
Curtea de Argeș	140	60	40	0	240
Făgăraș	60	80	20	0	160
Măneciu Ungureni	0	0	60	140	200
Bistrița	0	20	20	160	200
	300	150	140	300	

Din acest tabel reiese că singurul centru a cărui cerere nu a fost satisfăcută în totalitatea ei este centrul x_3^*.

Cu toate că mai există un disponibil de 40 t în centrul x_2 (Făgăraș), acestea nu mai pot fi transportate pentru satisfacerea centrului deficitar x_3^*, întrucît ruta (x_2, x_3^*) nu mai dispune de mijloace de transport. Pentru a ne convinge că într-adevăr cantitatea transportată este maximă, considerăm mulțimea

$$Y = \{x_3, x_4, x_2^*, x_3^*, x_4^*, Z,\}$$

Scanned with OKEN Scanner

cu tăietura

$$A_{\bar{Y}}\{=(x_1^*, Z);(x_1, x_2^*);(x_1, x_3^*);(x_2, x_2^*);(x_2, x_3^*);(x_0, x_3);(x_{0,4})x\}$$

și cu capacitatea

$$C(A_{\bar{Y}}) = C(x_1^*, Z) + C(x_1, x_2^*) + C(x_1, x_3^*) + C(x_2, x_2^*) + C(x_2, x_3^*) + C(x_0, x_3) + C(x_0, x_4) = 200 \text{ t} + 40 + 60 \text{ t} + 80 \text{ t} + 20 \text{ t} + 200 \text{ t} + 200 \text{ t} = 800 \text{ t}.$$

În exemplul tratat am făcut abstracție de costul transportului pe unitatea de produs sau de distanțele dintre producători și consumatori.

2°. Vom arăta în continuare cum rezolvăm o problemă de repartiție în situația în care sîntem obligați să ținem seama de aceste costuri sau distanțe.

În tabelul de mai jos sînt date cantitățile disponibile la centrele de producție, cantitățile necesare consumatorilor precum și costul transportului unității de produs de la fiecare producător la fiecare consumator.

A_i \ B_j	B_1	B_2	B_3	B_4	B_5	B_6	Disponibil
A_1	5	10	14	20	8	6	50
A_2	3	9	5	18	22	15	70
A_3	324	16	7	4	18	25	75
Necesar	20	30	60	10	50	25	195

Cerem să se facă o astfel de repartiție, încît toți consumatorii să fie satisfăcuți, în condițiile disponibilului dat, iar costul total al transportului să fie minim.

Arătăm mai întîi cum se construiește rețeaua corespunzătoare unei asemenea probleme:

a) considerăm o intrare x_0, care se leagă prin arce cu vîrfurile $A_{i(i=1,2,3)}$, corespunzătoare centrelor de expediție și cu capacități egale cu disponibilul, respectiv 50, 70, 75;

b) considerăm un vîrf Z (ieșirea din rețea), pe care îl legăm cu centrele consumatoare $B_{j(j=1,2,3,4,5,6)}$ prin arce orientate

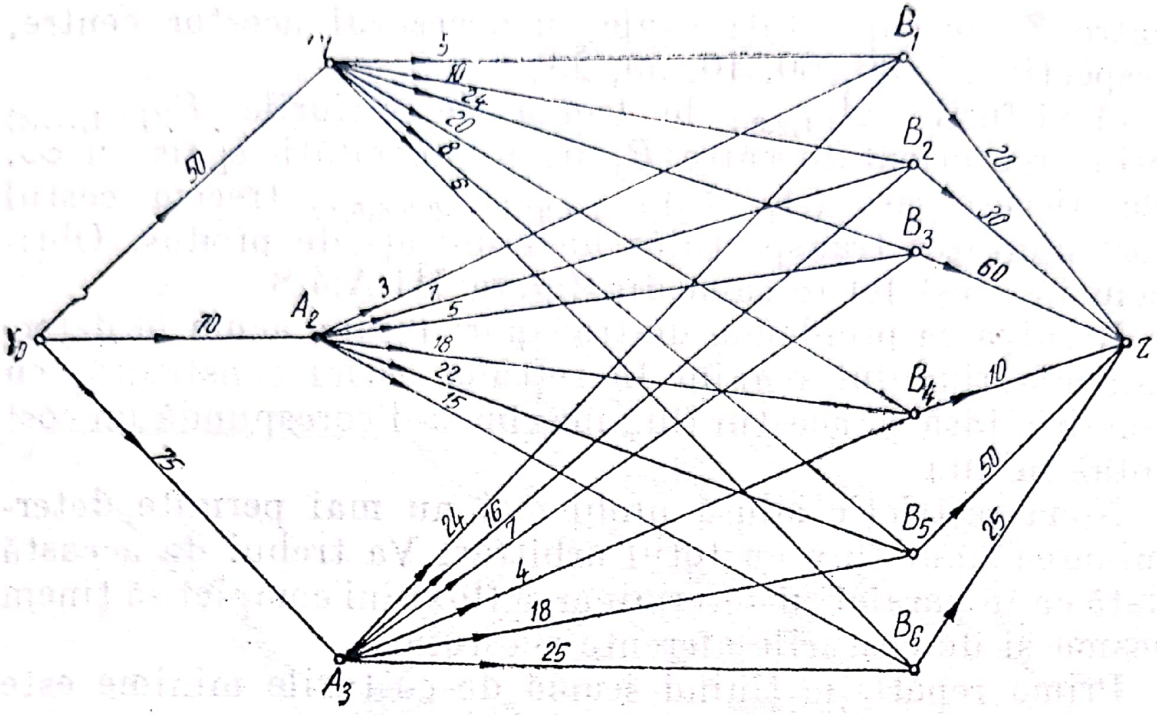


Fig. III.A.4.8

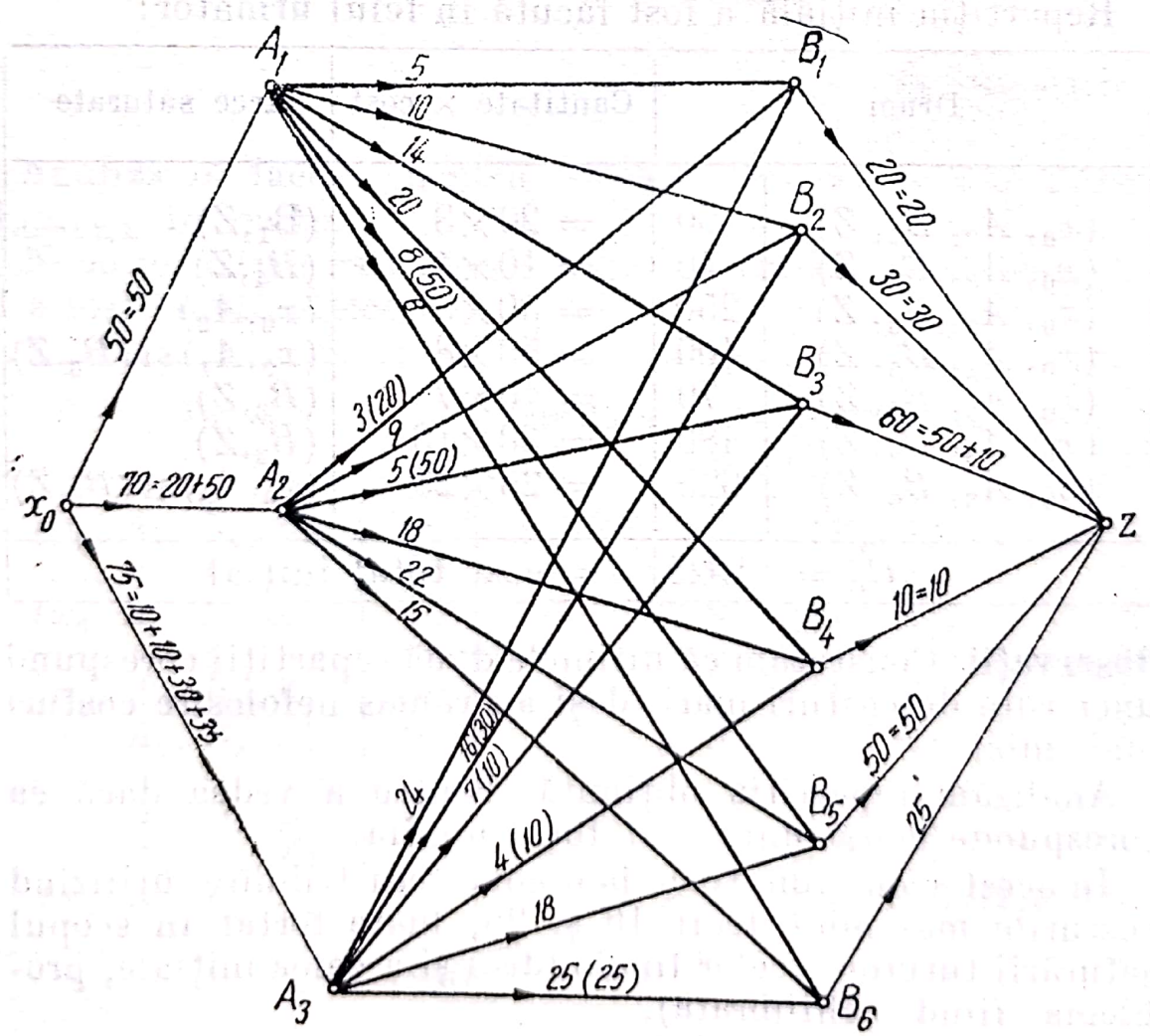


Fig. III.A.4.9

Scanned with OKEN Scanner

către Z, de capacități egale cu necesarul acestor centre, respectiv 20, 30, 60, 10, 50, 25;

c) vîrfurile $A_{i(1,2,3,)}$ le legăm de vîrfurile $B_{j(j=1,\ldots,6)}$ prin arce orientate către B_j și cu capacități egale cu ∞. Pe fiecare arc $(A_i, B_j)_{(i=1,2,3;\ j=1,2,3,4,5,6)}$ trecem costul corespunzător transportului unei unități de produs. Obținem în acest fel rețeaua din figura III.A.4.8.

Rezolvarea problemei de transport revine acum la determinarea fluxului maxim în rețeaua astfel construită, cu condiția însă ca acestui flux maxim să-i corespundă un cost total minim.

Noua restricție adusă problemei nu mai permite determinarea unui flux cu totul arbitrar. Va trebui de această dată ca în paralel cu determinarea fluxului complet să ținem seama și de costurile aferente acestuia.

Prima repartiție ținînd seamă de costurile minime este dată în graful din figura III.A.4.9.

Repartiția inițială a fost făcută în felul următor:

Drum		Cantitate × cost	Arce saturate
(x_0, A_2, B_1, Z)	60	$= 20\times3$	(B_1,Z)
(x_0, A_3, B_4, Z)	40	$= 10\times4$	(B_4,Z)
(x_0, A_2, B_3, Z)	250	$= 50\times5$	(x_0,A_2)
(x_0, A_1, B_5, Z)	400	$= 50\times8$	(x_0,A_1) și (B_5,Z)
(x_0, A_3, B_3, Z)	70	$= 10\times7$	(B_3,Z)
(x_0, A_3, B_2, Z)	480	$= 30\times16$	(B_2,Z)
(x_0, A_3, B_6, Z)	625	$= 25\times25$	(x_0,A_3) și (B_6,Z)
$C_0 =$	1925	= cost total inițial	

Observații: Constatăm că ultimele două repartiții corespund unor rute de costuri mari, deși au rămas nefolosite costuri mai mici.

Analizăm repartiția obținută, pentru a vedea dacă ea corespunde acum unui cost total minim.

În acest scop, vom trece la o nouă redistribuire, utilizînd costurile mai mici decît 16 și 25, luate forțat în scopul saturării tuturor arcelor finale (deci și a celor inițiale, problema fiind echilibrată).

Scanned with OKEN Scanner

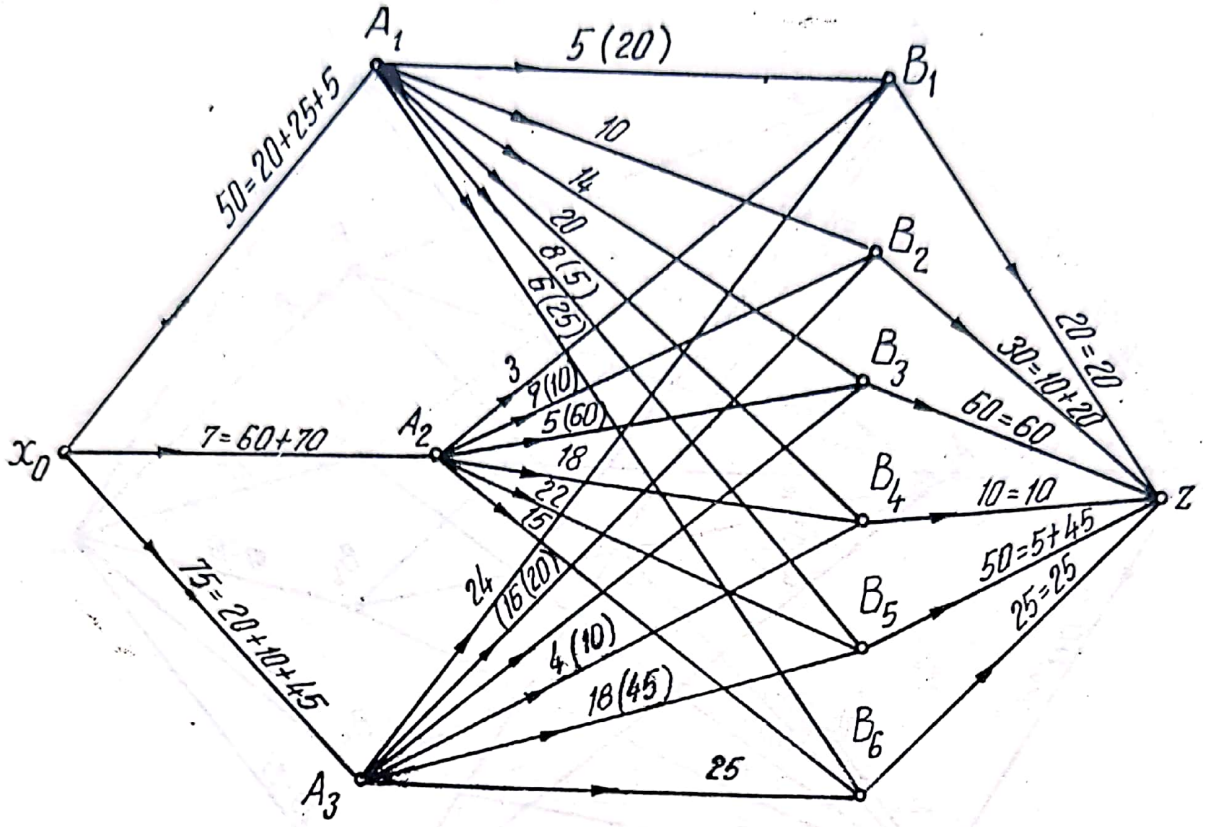


Fig. III.A.4.10

Analiza o facem pentru fiecare centru de expediție $A_{i(i=1,2,3)}$ în parte.

Noua repartiție este redată în graful din figura III.A.4.10 și a fost făcută astfel:

Drum		Cantitate × cost	Arce saturate
(x_0, A_1, B_1, Z)	100	$= 20 \times 5$	(B_1, Z)
(x_0, A_1, B_6, Z)	150	$= 25 \times 6$	(B_6, Z)
(x_0, A_1, B_5, Z)	40	$= 5 \times 8$	(x_0, A_1)
(x_0, A_2, B_3, Z)	300	$= 60 \times 5$	(B_3, Z)
(x_0, A_2, B_2, Z)	90	$= 10 \times 9$	(x_0, A_2)
(x_0, A_3, B_2, Z)	320	$= 20 \times 16$	(B_2, Z)
(x_0, A_3, B_4, Z)	40	$= 10 \times 4$	(B_4, Z)
(x_0, A_3, B_5, Z)	810	$= 45 \times 18$	(x_0, A_3) și (B_5, Z)

$C_1 = 1850$

Scanned with OKEN Scanner

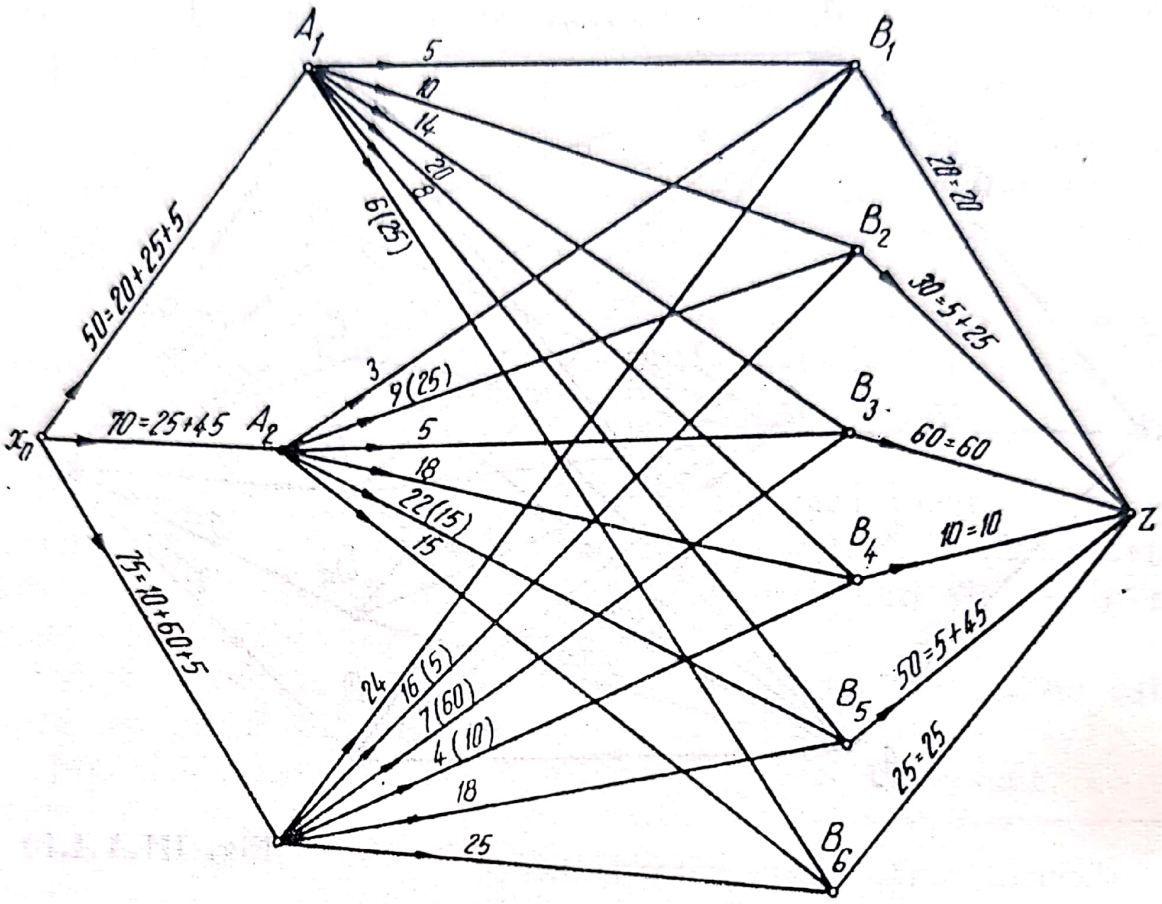


Fig. III.A.4.11

Comparînd costurile totale corespunzătoare celor două repartiţii, constatăm că cea de-a doua este mai bună.

În ceea ce priveşte vîrful A_3, vom avea următoarea repartiţie, redată în graful din figura III.A.4.11 şi analizată în tabelul care urmează.

Drum	Cantitate × cost	Arce saturate
(x_0, A_3, B_4, Z)	$10 \times 4 = 40$	(B_4, Z)
(x_0, A_3, B_3, Z)	$60 \times 7 = 420$	(B_3, Z)
(x_0, A_3, B_2, Z)	$5 \times 16 = 80$	(x_0, A_3)
(x_0, A_1, B_1, Z)	$20 \times 5 = 100$	(B_1, Z)
(x_0, A_1, B_6, Z)	$25 \times 6 = 150$	(B_6, Z)
(x_0, A_1, B_5, Z)	$5 \times 8 = 40$	(x_0, A_1)
(x_0, A_2, B_2, Z)	$25 \times 9 = 225$	(B_2, Z)
(x_0, A_2, B_5, Z)	$45 \times 22 = 990$	(B_5, Z) şi (x_0, A_2)

2045

Scanned with OKEN Scanner

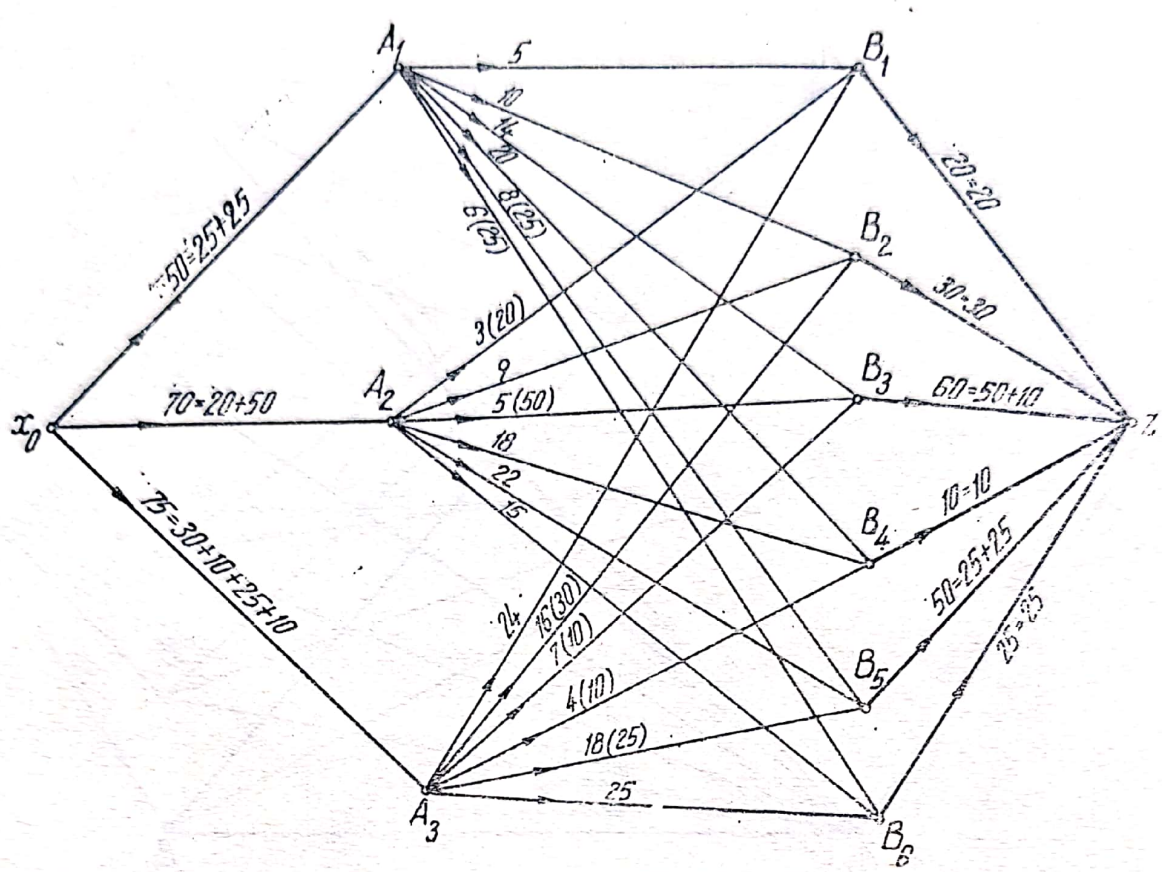


Fig. III.A.4.12

Varianta a treia conduce la un cost mai mare decît cel legat de varianta a doua, și ca atare, nu poate fi luată în considerație. Continuăm analiza legată de modificările de la centrul A_2 din varianta a II-a.

Drum	Cantitate × cost	Arce saturate
(x_0, A_2, A_1, Z) (x_6, A_2, B_3, Z)	$20 \times 3 = 60$ $50 \times 5 = 250$	(B_1, Z) (x_0, A_2)
(x_0, A_1, B_6, Z) (x_0, A_1, B_5, Z)	$25 \times 6 = 150$ $25 \times 8 = 200$	(B_6, Z) (x_0, A_1)
(x_0, A_3, B_2, Z) (x_0, A_3, B_4, Z) (x_0, A_3, B_5, Z) (x_0, A_3, B_3, Z)	$30 \times 16 = 480$ $10 \times 4 = 40$ $25 \times 18 = 450$ $10 \times 7 = 70$	(B_2, Z) (B_4, Z) (B_5, Z) $(x_0, A_3), (B_3, Z)$

1700

Scanned with OKEN Scanner

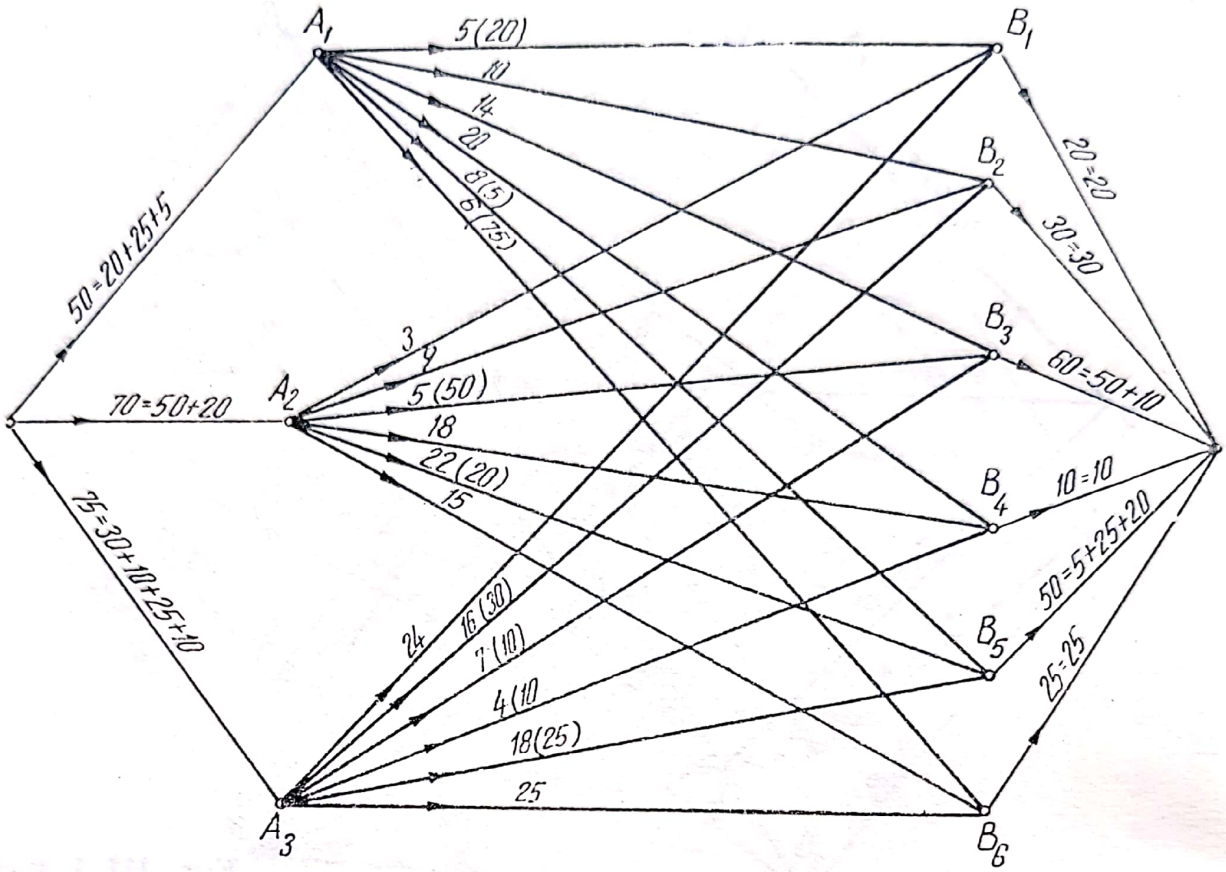


Fig. III.A.4.13

Ultima repartiție corespunde unui cost total mai mic în comparație cu varianta a doua.

Aplicăm grafului din figura III.A.4.12 același procedeu, obținînd graful din figura III.A.4.13.

Drum	Cantitate × cost	Arce saturate
(x_0, A_1, B_1, Z)	$20 \times 5 = 100$	(B_1, Z)
(x_0, A_1, B_6, Z)	$25 \times 6 = 150$	(B_6, Z)
(x_0, A_1, B_5, Z)	$5 \times 8 = 40$	(x_0, A_1)
(x_0, A_2, A_3, Z)	$50 \times 5 = 250$	
(x_0, A_2, B_5, Z)	$20 \times 22 = 440$	(x_0, A_2) și (B_5, Z)
(x_0, A_3, B_2, Z)	$30 \times 16 = 480$	(B_2, Z)
(x_0, A_3, B_4, Z)	$10 \times 4 = 40$	(B_4, Z)
(x_0, A_3, B_3, Z)	$10 \times 7 = 70$	(B_3, Z)

1570

Scanned with OKEN Scanner

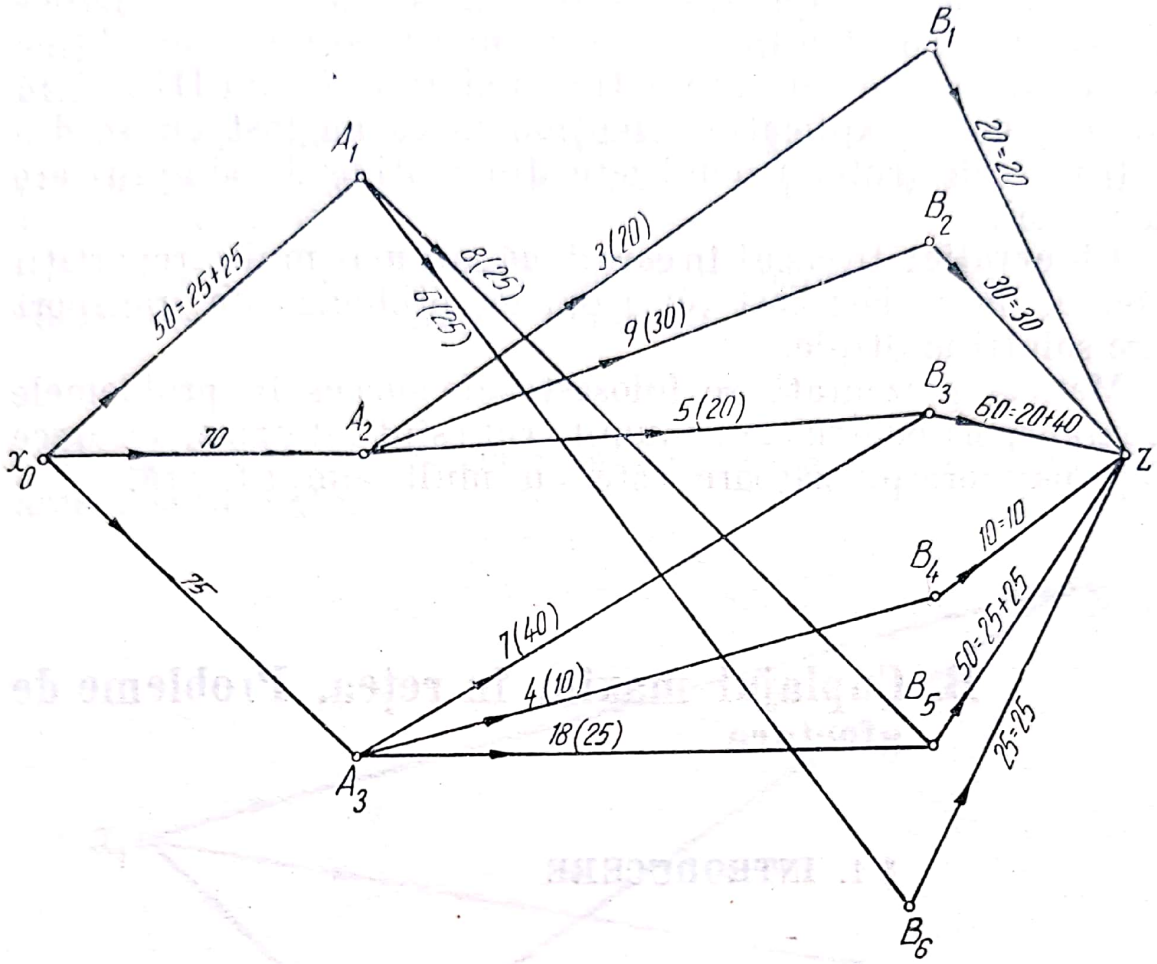


Fig. III.A.4.14

În graful din figura III.A.4.14 este redată soluția optimă a problemei.

Drum	Cantitate × cost	Arce saturate
(x_0, A_1, B_5, Z) (x_0, A_1, B_6, Z)	$25 \times 8 = 200$ $26 \times 6 = 150$	(B_6, Z) și (x_0, A_1)
(x_0, A_2, B_1, Z) (x_0, A_2, B_2, Z) (x_0, A_2, B_3, Z)	$20 \times 3 = 60$ $30 \times 9 = 270$ $20 \times 5 = 100$	(B_1, Z) (B_2, Z) (x_0, A_2)
(x_0, A_3, A_4, Z) (x_0, A_3, B_3, Z) (x_0, A_3, B_5, Z)	$10 \times 4 = 40$ $40 \times 7 = 280$ $25 \times 18 = 450$	(B_4, Z) (B_3, Z) (x_0, A_3) și (B_5, Z)

1550

Scanned with OKEN Scanner

Procedînd din aproape în aproape cu analizarea noilor redistribuiri pentru fiecare centru de expediere, vom obține în cele din urmă varianta optimă redată în figura III.A.4.14 și în tabelul explicativ. Menționăm că au fost omise din rețea arcele (rutele) nefolosite din motive de simplificare a rețelei.

Observație: În cazul în care două sau mai multe repartiții conduc la același cost total optim, problema de transport are soluții multiple.

Metoda prezentată se folosește cu succes în problemele de transport pentru care anumite rute sînt interzise, deoarece rețeaua corespunzătoare este cu mult simplificată.

B. Cuplajul maxim în rețea. Probleme de afectare

§ 1. INTRODUCERE

Strîns legată de problemele de repartiție este problema cuplajului maxim între două mulțimi disjuncte.

Noțiunea de cuplaj își găsește aplicații în sectoare importante ale producției. Dintre acestea, menționăm repartizarea lucrătorilor dintr-o întreprindere industrială pe locurile de muncă, repartizarea sarcinilor pe mașini, repartizarea personalului de deservire a transportului rutier, naval, aerian pe rutele deservite, în funcție de cerințele generale și cele individuale etc.

Fiecare din exemplele enumerate sînt alcătuite din două mulțimi disjuncte X și Y, în care prima conține elemente ce urmează a fi repartizate (lucrători, personal, sarcini), iar cea de a doua, elemente ce așteaptă să fie deservite (locuri de muncă, rute, mașini).

Dacă ne referim numai la primul exemplu, fiecăruia din lucrătorii mulțimii X i s-ar putea repartiza unul sau mai multe locuri de muncă, în funcție de calificare sau de alte criterii.

Scanned with OKEN Scanner

Evident că orice repartiție trebuie să conducă în final la asigurarea fiecărui loc de muncă cu un lucrător și numai unul.

Din punct de vedere matematic, exemplele considerate conduc la o aplicație F, definită pe mulțimea X cu valori în mulțimea Y.

Tripletul format din mulțimea X, mulțimea Y și aplicația F constituie un graf simplu notat cu

$$G = (X, Y, F).$$

Pe graful G, repartiția finală conduce la o mulțime de arce neadiacente.

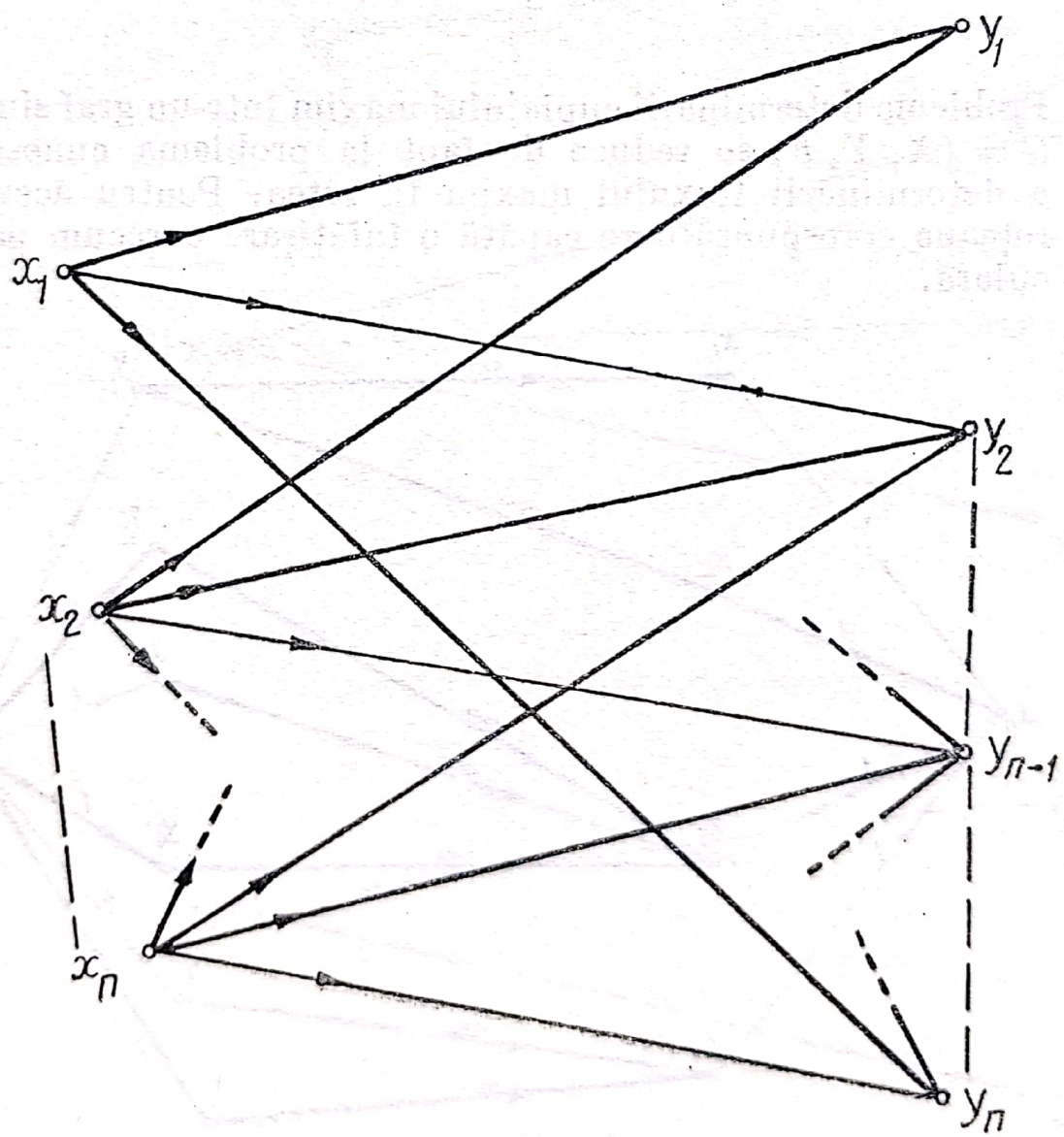


Fig. III.B.1.1

Scanned with OKEN Scanner

O astfel de mulțime formată din arce neadiacente, aparținînd grafului $G = (X, Y, F)$, poartă numele de *cuplaj*.

În același graf putem realiza mai multe cuplaje. Ținînd însă seama de cerințele practice, se impune alegerea acelui cuplaj care să conțină un număr maxim de arce.

Forma generală a grafului atașat unei asemenea probleme este redată în figura III.B.1.1.

Pe figură se poate citi cuplajul format din arcele

$$(x_1, y_2), (x_2, y_{n-1}) \ldots (x_n, y_n)$$

§ 2. CUPLAJUL MAXIM

Problema determinării cuplajului maxim într-un graf simplu $G = (X, Y, F)$ se reduce de fapt la problema cunoscută a determinării fluxului maxim în rețea. Pentru aceasta, rețeaua corespunzătoare capătă o înfățișare oarecum particulară.

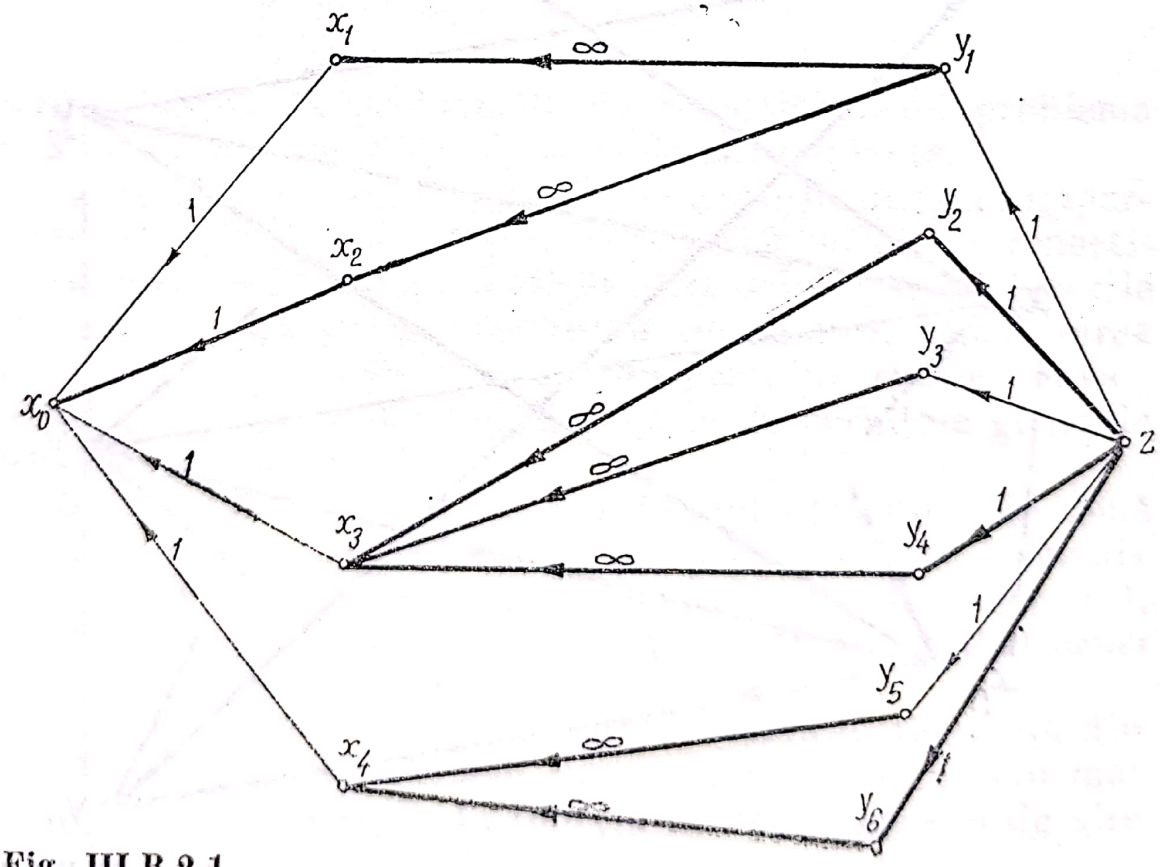


Fig. III,B,2,1

Scanned with OKEN Scanner

Considerăm vîrful x_0 ce devine de această dată ieșirea din rețea, care se leagă cu fiecare vîrf al mulțimii X, prin arce incidente spre interior lui x_0 și de capacități egale cu 1, și un vîrf z (intrarea în rețea) legat de toate vîrfurile mulțimii Y cu arce incidente spre exterior lui și de capacități egale cu unitatea. Arcele care leagă elementele mulțimii X cu elementele mulțimii Y sînt incidente spre interior mulțimii y

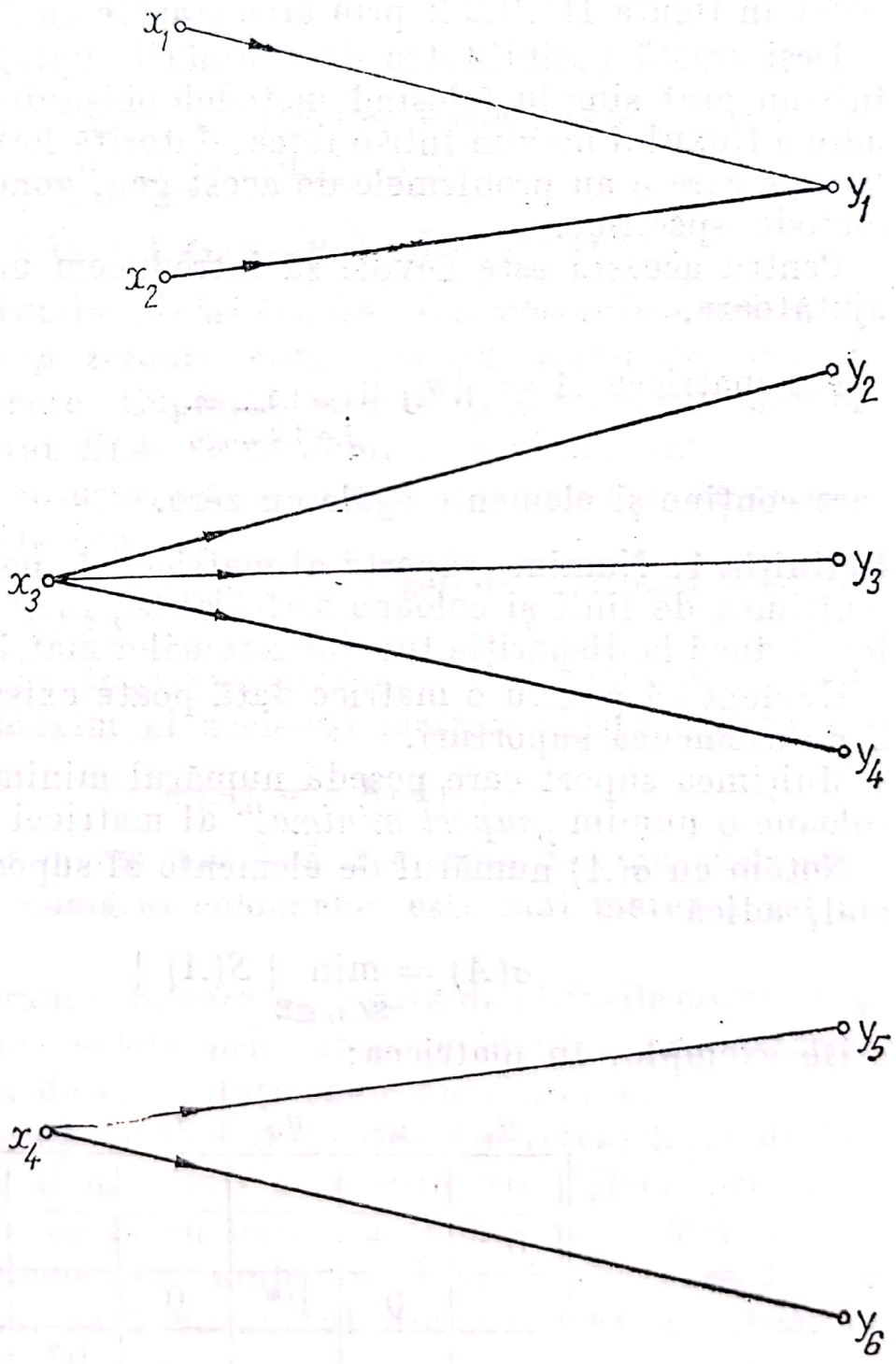


Fig. III.B.2.2

Scanned with OKEN Scanner

sînt incidente spre interior mulțimii X și au capacități nelimitate.

În rețeaua astfel construită (figura III.B.2.1) se determină fluxul maxim în x_0, care corespunde cuplajului maxim ce se poate realiza între X și Y.

Aplicînd algoritmul Ford-Fullkerson, observăm că fluxul maxim este de asemenea egal cu 3. Cuplajul maxim este redat în figura III.B.2.2 prin arce trasate cu linie subțire.

Deși există posibilitatea determinării cuplajului maxim într-un graf simplu, folosind metodele obișnuite de determinare a fluxului maxim într-o rețea, datorită formei particulare pe care o au problemele de acest gen, vom prezenta și metode specifice.

Pentru aceasta este nevoie să introducem unele noțiuni ajutătoare.

Fie matricea $A = \| a_{ij} \|_{\substack{i=1,2,\ldots m. \\ j=1,2,\ldots, n,}}$

care conține și elemente egale cu zero.

Definiția 1. Numim „suport" al matricei A, notat cu $S(A)$ mulțimea de linii și coloane astfel alese, încît suprimarea lor să ducă la dispariția tuturor zerourilor matricei A.

Evident că pentru o matrice dată poate exista o familie Σ de asemenea suporturi.

Mulțimea suport care posedă numărul minim de linii și coloane o numim „*suport minimal*" al matricei A.

Notăm cu $\sigma(A)$ numărul de elemente al suportului minimal, adică

$$\sigma(A) = \min_{S(A) \in \Sigma} | S(A) |$$

De exemplu, în matricea:

$A =$

	y_1	y_2	y_3	y_4	y_5	y_6
x_1	0*					
x_2	0					
x_3		0	0*	0		
x_4					0*	0

Scanned with OKEN Scanner

putem considera suporturile:

$$S^{(1)}(A) = \{ x_1, x_2, x_3, x_4 \}$$
$$S^{(2)}(A) = \{ y_1, y_2, y_3, y_4, y_5, y_6 \},$$
$$S^{(3)}(A) = \{ x_1, x_2, x_3, y_5, y_6 \},$$
$$S^{(4)}(A) = \{ x_2, x_4, y_1, y_5, y_6 \},$$
$$S^{(5)}(A) = \{ x_3, x_4, y_1 \}$$

etc.

Din mulțimea Σ a suporturilor posibile ale matricei A observăm că suportul minimal este

$$S^{(5)}(A) = \{ x_3, x_4, y_1 \}, \text{ deci } \sigma(A) = 3$$

Definiția 2. Numim *cuplaj de* p *linii și* p *coloane în matricea A* mulțimea de p zerouri situate la intersecții de linii și coloane distincte. Cuplajul care conține numărul maxim de zerouri notat $K(A)$ se va numi „cuplaj maxim".

Constatăm cu ușurință că în matricea considerată cuplajul maxim este egal cu 3.

Unul din aceste cuplaje este de zerourile situate la intersecțiile (x_1, y_1); (x_3, y_3); (x_4, y_5).

Arătăm acum că în general între suportul minim al matricei A și cuplajul maxim al aceleiași matrice există egalitatea.

$$\sigma(A) = |K(A)|$$

Mai întîi, asociem matricei A o rețea de transport, presupunînd că numărul coloanelor este mai mare ca cel al liniilor.

a) Considerăm o intrare x_0, legată de vîrfurile mulțimii y (mulțimea coloanelor) prin arce incidente spre exterior vîrfului x_0 și de capacități egale cu unitatea.

b) Considerăm un vîrf Z (ieșirea din rețea) legat de fiecare element al mulțimii x (mulțimea liniilor), prin arce de capacitate egală cu unitatea și orientate către Z.

c) Între elementele mulțimii Y și X vor exista arce (y_j, x_i) numai dacă elementul din matricea A, situat la intersecția liniei x_i, cu coloana y_j, este egal cu zero.

Scanned with OKEN Scanner

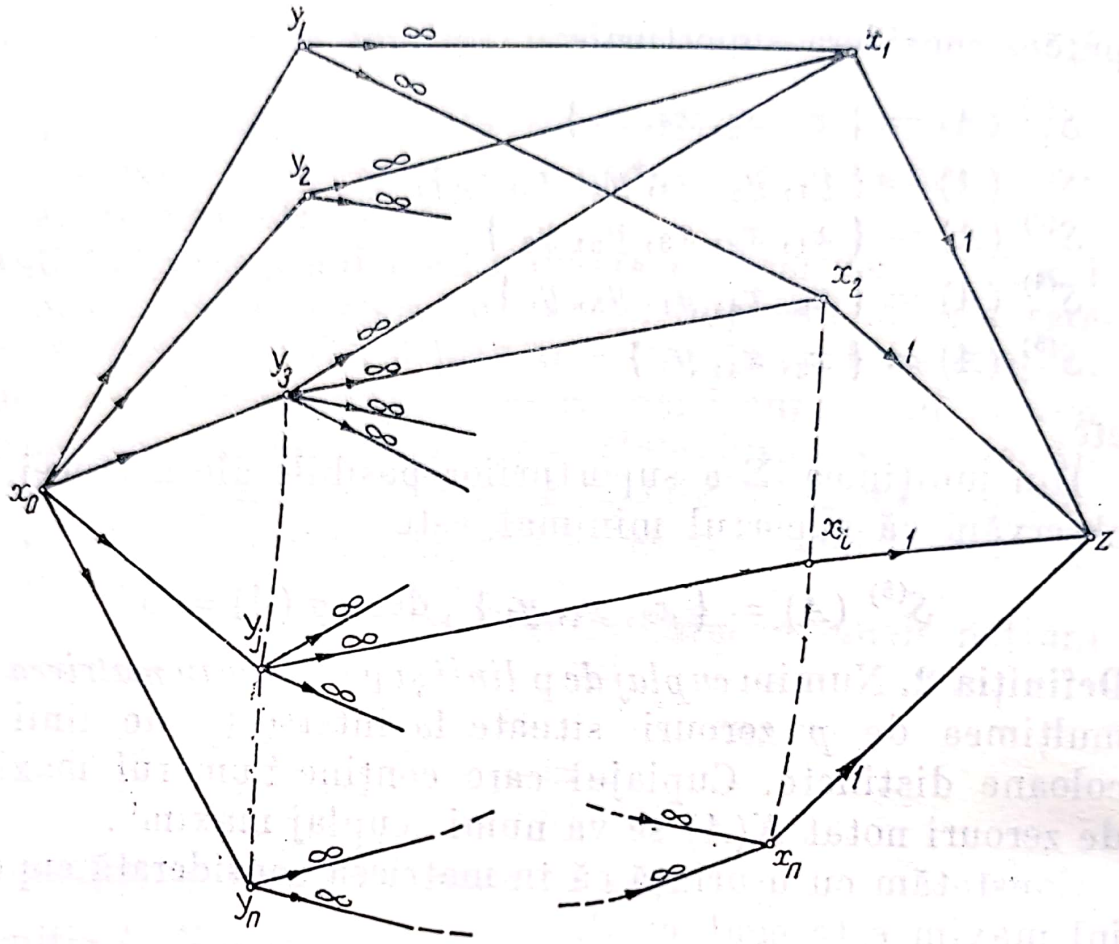


Fig. III.B.2.3

Forma generală a unei rețele de transport asociată unei matrice pe baza celor arătate este prezentată în figura III.B.2.3.

Fie $X = \{ x_1, x_2, \ldots, x_n \}$ mulțimea liniilor,

$Y = \{ y_1, y_2, \ldots, y_m \}$ mulțimea coloanelor,

$$S(A) = \{ x_{i_1}, x_{i_2}, \ldots, x_{i_p}, y_{j_1}, y_{j_1}, \ldots y_{j_q} \}.$$

Introducem notațiile:

$$X^+ = \{ x_i \in X \mid x_i \in S(A) \},$$

$$X^- = \{ x_i \in X \mid x_i \notin S(A) \},$$

$$Y^+ = \{ y_j \in Y \mid y_j \in S(A) \},$$

$$Y^- = \{ y_j \in Y \mid y_j \notin S(A) \}.$$

Scanned with OKEN Scanner

În rețeaua asociată nu există nici un zero care să lege elementele mulțimii X^- cu elementele mulțimii Y^-.

Arătăm că oricărui suport îi corespunde o tăietură de capacitate finită.

Fie acum T mulțimea care conține ieșirea din rețea Z, elementele mulțimii X^- și elementele mulțimii Y^+.

Cu ajutorul mulțimii T, definim o tăietură A_T^- în componența căreia intră:

a) Arce care au extremitatea finală în Z și extremitatea inițială în X^+, cu capacitatea egală cu unitatea.

b) Arce care au extremitatea inițială în x_0 și extremitatea finală în Y^+.

Dacă $|X^+| = p$ și $|Y^+| = q$, atunci avem $C(A_T^-) = p + q$, adică

$$|S(A)| = C(A_T^-).$$

Arătăm în continuare că și reciproca este adevărată, adică:

Oricărei tăieturi de capacitate finită îi corespunde un suport.

Într-adevăr, oricare ar fi tăietura de capacitate finită A_T^-, mulțimea T nu poate să conțină decît vîrfuri din X^+ sau Y^+, pentru că în caz contrar, $C(A_T^-) = \infty$, ceea ce contrazice ipoteza.

Matricea formată cu liniile X^- și Y^- nu conține nici un zero și ca atare X^+ și Y^+ constituie un suport.

Dacă $A_{T_0}^-$ este o tăietură minimă, ei îi corespunde un suport minim $S^{(0)}(A)$, de unde

egalitatea $|S^{(0)}(A)| = \sigma(A) = C(A_{T_0}^-)$.

Pe de altă parte, capacitatea minimă a tăieturii este egală cu fluxul maxim în rețea, și prin urmare:

$$\sigma(A) = C(A_{T_0}^-) = f_{\max}.$$

Cum însă fluxul maxim în rețea corespunde numărului maxim de cercuri ale cuplajului, avem relația finală

$$\sigma(A) = |K(A)|.$$

Scanned with OKEN Scanner

Din cele prezentate pînă acum rezultă posibilitatea determinării cuplajului maxim determinînd suportul minim al matricei A ale cărei elemente reprezintă posibilitățile de repartiție.

Pentru aceasta, înlocuim matricea A cu complementara ei, în care facem operațiile.

a) Încadrăm un zero de pe linia cu cele mai puține zerouri și suprimăm celelalte zerouri de pe linia și coloana la intersecția cărora este situat zeroul încadrat.

b) Încadrăm din nou un zero din linia rămasă cu cele mai puține zerouri și suprimăm zerourile situate pe linia și coloana zeroului încadrat.

Continuăm în acest fel, pînă cînd toate zerourile matricei sînt sau încadrate, sau tăiate.

Mulțimea de zerouri încadrate formează suportul minim al matricei sau încă, cuplajul maxim.

Exemplu. Fie 5 muncitori $\{x_1, x_2, x_3, x_4, x_5\}$ ce trebuie repartizați pe 5 locuri de muncă $\{y_1, y_2, y_3, y_4, y_5\}$.

Posibilitățile de repartizare a muncitorilor sînt redate în graful simplu din figura III.B.2.5.

Avîndu-se în vedere posibilitățile unuia și aceluiași muncitor de a lucra pe mai multe posturi, să se facă o repartiție de așa manieră, încît fiecărui muncitor să-i corespundă un post și numai unul și fiecare post să fie ocupat de un muncitor și numai unul.

Aceasta, pentru a determina cuplajul maxim în graful din figura III.B.2.4.

Matricea asociată grafului este:

$$A = \begin{array}{c|ccccc} & y_1 & y_2 & y_3 & y_4 & y_5 \\ \hline x_1 & 1 & 1 & 0 & 0 & 0 \\ x_2 & 1 & 0 & 1 & 0 & 1 \\ x_3 & 0 & 1 & 1 & 1 & 0 \\ x_4 & 0 & 0 & 0 & 1 & 0 \\ x_5 & 1 & 0 & 1 & 1 & 1 \end{array}$$

Scanned with OKEN Scanner

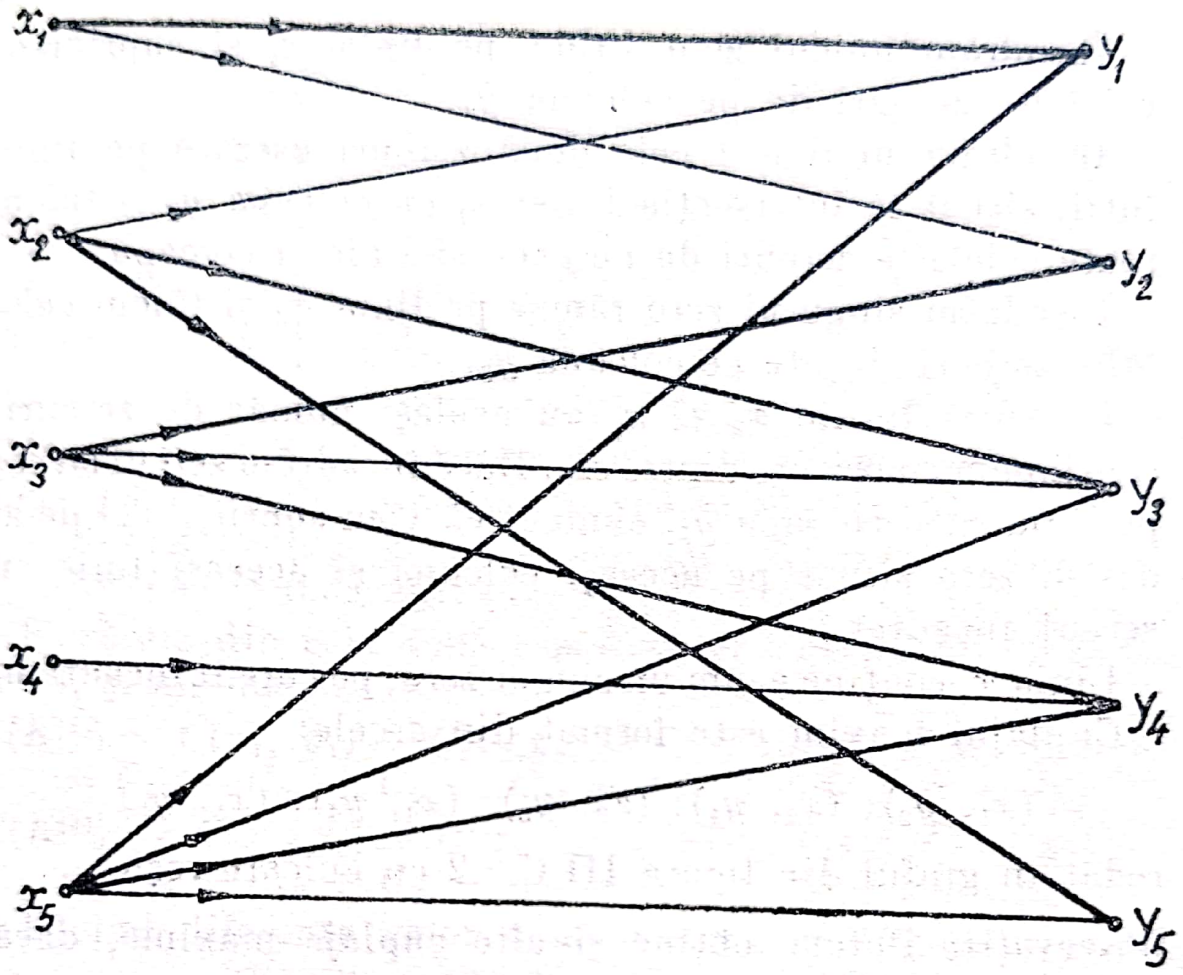


Fig. III.B.2.4

Complementara A' a matricei A este:

$$A' = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{ccccc} y_1 & y_2 & y_3 & y_4 & y_5 \end{array} \\ \left\| \begin{array}{ccccc} \not{0} & \boxed{0} & 1 & 1 & 1 \\ \boxed{0} & 1 & \not{0} & 1 & \not{0} \\ 1 & \not{0} & \boxed{0} & \not{0} & 1 \\ 1 & 1 & 1 & \boxed{0} & 1 \\ \not{0} & 1 & \not{0} & \not{0} & \boxed{0} \end{array} \right\| \end{array}$$

Scanned with OKEN Scanner

Încadrăm unicul zero situat pe linia x_4 și suprimăm celelalte zerouri de pe coloana y_4.

Încadrăm unul din cele două zerouri așezate pe linia întîi, situat la intersecția liniei x_1 cu coloana y_2 și tăiem toate celelalte zerouri de pe aceeași linie și coloană.

Încadrăm singurul zero rămas pe linia x_3 și tăiem celelalte zerouri situate pe coloana y_3.

Deoarece liniile x_2 și x_5 au același număr de zerouri, continuăm cu oricare dintre ele. Dacă încadrăm zerul situat pe linia x_2 și coloana y_1, evident că vom suprima cel de-al doilea zero situat pe aceeași coloană și aceeași linie cu zeroul încadrat.

Linia x_5 conține acum un singur zero, pe care îl încadrăm.

Cuplajul maxim este format din arcele:

$$(x_1, y_2);\ (x_2, y_1);\ (x_3, y_3);\ (x_4, y_4);\ (x_5, y_5)$$

redat în graful din figura III.C.2.2 cu culoare roșie.

Observații: Putem obține și alte cuplaje maxime, dacă alegerea zerourilor este alta.

§3. CUPLAJUL MAXIM LEGAT DE COSTUL TOTAL AL REPARTIZĂRILOR

În cele prezentate pînă acum am presupus că repartițiile muncitorilor x_i pe posturile y_j nu implică salarizări diferite.

Evident că, din punct de vedere economic, existența unor cheltuieli diferite pentru fiecare repartiție (x_i, y_j) impune o dată cu determinarea cuplajului maxim să se obțină și costul total minim, legat de aceste repartizări.

Considerăm matricea A, asociată unui graf $G = (X, Y, F)$, în care elementele sale arată costurile corespunzătoare repartițiilor.

Scanned with OKEN Scanner

$$A = \begin{array}{c|cccc} & y_1 & y_2 & y_3 & y_4 \\ \hline x_1 & 2 & 5 & 1 & 7 \\ x_2 & 4 & 6 & 3 & 2 \\ x_3 & 7 & 9 & 7 & 3 \\ x_4 & 2 & 1 & 3 & 5 \end{array}$$

În matricea A, să luăm în considerație două cuplaje maxime arbitrare din punctul de vedere al costurilor:

$$K^{(1)}(A) = \{(x_1, y_1); (x_2, x_3); (x_3, y_2); (x_4, y_4)\},$$
$$K^{(2)}(A) = \{(x_1, y_2); (x_2, y_3); (x_3, y_4); (x_4, y_1)\}.$$

Fiecăreia din cele două repartiții îi corespunde un cost total, și anume,

$$C(K^{(1)}) = C(x_1, y_1) + C(x_2, y_3) + C(x_3, y_2) + C(x_4, y_4) =$$
$$= C_{11} + C_{23} + C_{32} + C_{44} = 2 + 3 + 9 + 5 = 19,$$
$$C(K^{(2)}) = C(x_1, y_2) + C(x_2, y_3) + C(x_3, y_4) + C(x_4, y_1) =$$
$$= C_{12} + C_{23} + C_{34} + C_{41} = 5 + 3 + 3 + 2 = 13.$$

Făcînd diferența celor două costuri, se obține:

$$C(K^{(1)}) - C(K^{(2)}) = 19 - 13 = 6.$$

Să arătăm că această diferență a costurilor celor două cuplaje nu este afectată, dacă în matricea dată scădem minoranții liniilor din elementele corespunzătoare și minoranții coloanelor din elementele coloanelor matricei $\tilde{A}$ obținute:

$$A = \begin{array}{c|cccc|c} & y_1 & y_2 & y_3 & y_4 & \alpha_i \text{ minoranții liniilor} \\ \hline x_1 & 2 & 5 & 1 & 7 & 1 \\ x_2 & 4 & 6 & 3 & 2 & 2 \\ x_3 & 7 & 9 & 7 & 3 & 3 \\ x_4 & 2 & 1 & 3 & 5 & 1 \end{array} \qquad A_1 = \begin{array}{c|cccc} & y_1 & y_2 & y_3 & y_4 \\ \hline x_1 & 1 & 4 & 0 & 6 \\ x_2 & 2 & 4 & 1 & 0 \\ x_3 & 4 & 6 & 4 & 0 \\ x_4 & 1 & 0 & 2 & 4 \\ \hline \beta_j \text{ minoranții coloanelor} & 1 & 0 & 0 & 0 \end{array}$$

$$\widetilde{A} = \begin{array}{c|cccc} & y_1 & y_2 & y_3 & y_4 \\ \hline x_1 & 0 & 4 & 0 & 6 \\ x_2 & 1 & 4 & 1 & 0 \\ x_3 & 3 & 6 & 4 & 0 \\ x_4 & 0 & 0 & 2 & 4 \end{array}$$

Costurile corespunzătoare cuplajelor considerate, ținînd seama de elementele matricei $\widetilde{A}$ sînt:

$$\widetilde{C}(K^{(1)}) = \widetilde{C}_{11} + \widetilde{C}_{23} + \widetilde{C}_{32} + \widetilde{C}_{44} = 0 + 1 + 6 + 4 = 11,$$

$$\widetilde{C}(K^{(2)}) = \widetilde{C}_{12} + \widetilde{C}_{23} + \widetilde{C}_{34} + \widetilde{C}_{41} = 4 + 1 + 0 + 0 = 5.$$

Diferența

$$\widetilde{C}(K^{(1)}) - \widetilde{C}(K^{(2)}) = 11 - 5 = 6 = \widetilde{C}(K^{(1)}) - \widetilde{C}(K^{(2)})$$

În general, considerăm matricea A a costurilor afectate repartițiilor posibile $A = \| C_{ij} \|$ $i, j = 1,2,\ldots, n$ și construim matricea $\widetilde{A}$ ale cărei elemente $\widetilde{C}_{ij}$ sînt:

$$\widetilde{C}_{ij} = C_{ij} - (\alpha_i + \beta_i)\ i, j = 1,\ldots,n,$$

unde

$$\alpha_i = \min_j C_{ij} \text{ și } \beta_j = \min_i (C_{ij} - \alpha_i).$$

Fie $K^{(1)}$ și $K^{(2)}$ două cuplaje maxime și $C(K^{(1)})$, $C(K^{(2)})$ costurile corespunzătoare din matricea A, iar $C(K^{(1)})$ și $C(K^{(2)})$ costurile corespunzătoare cuplajelor ținînd seama de elementele matricei $\widetilde{A}$. Arătăm că:

$$C(K^{(1)}) - C(K^{(2)}) = \widetilde{C}(K^{(1)}) - \widetilde{C}(K^{(2)}).$$

Diferențele

$$C(K^{(1)}) - C(K^{(2)}) =$$

$$C(j_1 = + C_{2j_2} + \ldots + C_{nj_n}) - (C_{1k_1} + C_{2k_2} + \ldots + C_{nk_n}) =$$

$$= (C_{1j_1} - C_{1k_1}) + (C_{2j_2} - C_{2k_2}) + \ldots + (C_{nj_n} - C_{nk_n}),$$

unde $j_1, j_2, \ldots, j_n$ și $k_1, k_2, \ldots, k_n$ formează permutări ale numerelor $1,2,\ldots n$.

$$\widetilde{C}(K^{(1)}) - \widetilde{C}(K^{(2)}) = (\widetilde{C}_{1j_1} + \widetilde{C}_{2j_2} + \ldots + \widetilde{C}_{nj_n}) - (\widetilde{C}_{1k_1} +$$

$$+ \widetilde{C}_{2k_2} + \ldots + \widetilde{C}_{nk_n}) = (\widetilde{C}_{1j_1} - \widetilde{C}_{1k_1}) + (\widetilde{C}_{2j_2} - \widetilde{C}_{2k_2}) + \ldots$$

$$\ldots + (\widetilde{C}_{nj_n} - \widetilde{C}_{nk_n}) = \left\{\left[C_{1j_1} - (\alpha_1 + \beta_{j1})\right] + \left[C_{2j_2} - (\alpha_2 +\right.\right.$$

$$+ \beta_{j2}) + \ldots + \left[C_{nj_n} - (\alpha_n + \beta_{jn})\right]\right\} - \left\{\left[C_{1k_1} - (\alpha_1 +\right.\right.$$

$$+ \beta_{k1})\left.\right] + \left[C_{2k_2} - (\alpha_2 + \beta_{k2})\right] + \ldots + \left[(C_{nk_n} - (\alpha_n +\right.$$

$$+ \beta_{kn})\left.\right]\right\} = \sum_{i=1}^{n} C_{ij_i} - \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \beta_{ji} - \sum_{i=1}^{n} C_{ik_i} + \sum_{i=1}^{n} \beta_{ki} +$$

$$+ \sum_{i=1}^{n} \alpha_i = \sum_{i=1}^{n} C_{ij_i} - \sum_{i=1}^{n} C_{ik_i} = C(K^{(1)}) - C(K^{(2)}).$$

Deoarece suma $\sum_{i=1}^{n} \beta_{ji}$ nu diferă de suma $\sum_{i=1}^{n} \beta_{ki}$ decît prin ordinea termenilor, rezultă că diferența

$$\sum_{i=1}^{n} \beta_{ji} - \sum_{i=1}^{n} \beta_{ki} = 0$$

și deci relația de mai sus este justificată.

Dacă luăm în considerație toate cuplajele maxime posibile în graful $G = (X, Y, F)$, atunci costurile corespunzătoare lor admit o margine inferioară.

Am considerat numerele,

$$\alpha_i = \min_j C_{ij};\ i = 1,2,\ldots,n, \tag{1}$$

$$\beta_j = \min_j (C_{ij} - \alpha_i);\ j = 1,2,\ldots,n. \tag{2}$$

Din relația (2) rezultă inegalitatea:

$$C_{ij} - \alpha_i \geqslant \beta_j \text{ (oricare ar fi } i, j, = 1,2,\ldots, n),$$

de unde

$$C_{ij} \geqslant \alpha_i + \beta_j \,. \tag{3}$$

Oricare ar fi cuplajul,

$$K(A) = (x_1, y_{j_1}), (x_2, y_{j_2}) \ldots, (x_n, y_{j_n})$$

avem relația

$$C(K) = C_{1j_1} + C_{2j_2} + \ldots + C_{nj_n} \geqslant (\alpha_1 + \beta_{j_1}) + (\alpha_2 + \beta_{j_2}) + \ldots + (\alpha_n + \beta_{j_n}) = \sum_{i=1}^{n} \alpha_i + \sum_{r=1}^{n} \beta_{j_r} = \sum_{i=1}^{n} \alpha_i + \sum_{j=1}^{n} \beta_j.$$

Indicii $j_1, j_2,\ldots, j_n$ reprezintă o permutare a numerelor $1,2,\ldots,n$,

$$C(K) \geqslant \sum_{i=1}^{n} \alpha_i + \sum_{j=1}^{n} \beta_j. \tag{4}$$

Vedem acum în ce caz există egalitate strictă între costul total al cuplajului și marginea inferioară $\sum_i \alpha_i + \sum_j \beta_j$.

Fie K un cuplaj al cărui cost total verifică egalitatea

$$C(\bar{K}) = \sum_{i=1}^{n} \alpha_i + \sum_{j=1}^{n} \beta_j.$$

Dar ținînd seama de faptul că

$$C(\bar{K}) = \sum_{i=1}^{n} C_{ij_i} = \sum_{i} (\alpha_i + \beta_{j_i}),$$

rezultă

$$\bar{C}_{ij_i} = \alpha_i + \beta_{i_i} \qquad (i = 1,\ldots,n)$$

Scanned with OKEN Scanner

și deci

$$\tilde{\tilde{C}}_{ij_i} = \tilde{C}_{ij_i} - (\alpha_i + \beta_{ji}) = 0. \qquad (5)$$

Observații: 1) Din relația (5) deducem că realizarea cuplajului maxim cu cost total minim se face pe căsuțele cu zerouri situate la intersecții de linii și coloane distincte în matricea $\tilde{A}$.

2) S-ar putea întîmpla ca $C(\bar{K})$ minim să nu corespundă unui cuplaj maxim, adică să nu conțină „n" zerouri. În acest caz, luăm drept matrice inițială matricea $\tilde{A}$, căreia i se aplică aceleași operații ca mai sus, pînă obținem cuplajul maxim format din n zerouri.

Să reluăm matricea $\tilde{A}$ și să-i determinăm suportul minim după cum s-a mai arătat.

$$S = \begin{array}{c|cccc} & y_1 & y_2 & y_3 & y_4 \\ \hline x_1 & \not{0} & 4 & \underline{|0|} & 6 \\ x_2 & 1 & 4 & 1 & \underline{|0|} \\ x_3 & 3 & 6 & 4 & \not{0} \\ x_4 & \underline{|0|} & \not{0} & 2 & 4 \end{array}$$

Avem: $S(A) = \{x_1, x_4, y_4\}$ cu $\sigma(A) = 3$.

Cuplajul corespunzător suportului minim din matricea $\tilde{A}$ este

$$K(\tilde{A}) = \{(x_1, y_3); (x_2, y_4); (x_4, y_1)\},$$

căruia în matricea inițială îi corespunde costul total:

$$C(K(A)) = C_{13} + C_{24} + C_{41} = 1 + 2 + 2 = 5.$$

Însă cuplajul $K(A)$ nu conține patru zerouri; în această situație, căutăm cuplajul maxim trecînd la modificarea valorilor α_i și β_j.

Considerăm pentru aceasta matricea formată din liniile și coloanele matricei $\widetilde{A}$, care nu aparțin suportului

$$A_1 = \begin{array}{c|ccc} & y_1 & y_2 & y_3 \\ \hline x_2 & 1 & 4 & 1 \\ x_3 & 3 & 6 & 4 \end{array}$$

Cel mai mic element al matricei $\widetilde{A}_1$ este egal cu 1. Modificăm valorile α_i și β_j în sensul următor:

$$\alpha_i' = \begin{cases} \alpha_i + r, \text{ dacă linia } x_i \text{ nu aparține suportului} \\ \alpha_i \quad , \text{ dacă linia } x_i \text{ aparține suportului,} \end{cases}$$

$$\beta_j' = \begin{cases} (\beta_j - r, \text{ dacă coloana } y_j \text{ aparține suportului} \\ \beta_j \quad , \text{ dacă coloana } y_j \text{ nu aparține suportului.} \end{cases}$$

Noile valori α'_i și β'_j sînt:

$$\alpha_1' = \alpha_1 = 1 \qquad \beta_1' = \beta_1 = 1$$
$$\alpha_2' = \alpha_2 + r = 2 + 1 = 3 \qquad \beta_2' = \beta_2 = 0$$
$$\alpha_3' = \alpha_3 + r = 3 + 1 = 4 \qquad \beta_3' = \beta_3 = 0$$
$$\alpha_4' = \alpha_4 = 1 \qquad \beta_4' = \beta_4 - r = 0 - 1 = 1.$$

Matricea $\widetilde{\widetilde{A}} = \|C_{ij} - (\alpha_i' + \beta_j')\| \; i,j = 1,2,3,4$ este următoarea:

$$\widetilde{\widetilde{A}} = \begin{array}{c|cccc|c} & y_1 & y_2 & y_3 & y_4 & \alpha'_i \\ \hline x_1 & 0 & 4 & \boxed{0} & 7 & 1 \\ x_2 & \boxed{0} & 3 & 0 & 0 & 3 \\ x_3 & 2 & 5 & 3 & \boxed{0} & 4 \\ x_4 & 0 & \boxed{0} & 2 & 5 & 1 \\ \hline \beta'_j & 1 & 0 & 0 & -1 & \end{array}$$

Suportul matricei $\tilde{\tilde{A}}$ este deci

$$S(\tilde{\tilde{A}}) = \{ x_1, x_2, x_3, x_4 \}$$

sau

$$S(\tilde{\tilde{A}}) = \{ y_1, y_2, y_3, y_4 \},$$

unde $\sigma(A) = 4$.

Cuplajul corespunzător suportului considerat este

$$K(\tilde{\tilde{A}}) = \{ (x_1, y_3); (x_2, y_1); (x_3, x_4); (x_4, y_2) \},$$

cu costul total

$$C(\bar{K}) = \bar{C}_{13} + \bar{C}_{21} + \bar{C}_{34} + \bar{C}_{42} = 1 + 4 + 3 + 1 = 9.$$

Cum însă K este cuplajul maxim realizat în matricea A și întrucît

$$C(K) = \sum_i \alpha'_i + \sum_j \beta'_j,$$

rezultă că el corespunde costului total minim.

Făcînd o analiză a celor prezentate, putem deduce unele reguli practice de calcul, care să conducă la determinarea cuplajului maxim corespunzător unui cost total minim.

Schematic, matricea $\tilde{A}$ are următorul aspect:

$\tilde{A} =$

	y_1	y_2	y_3	y_4
x_1	$C_{11}-(\alpha_1+\beta_1)$ $= \not{0}$	$C_{12}-(\alpha_1+\beta_2)$	$C_{13}-(\alpha_1+\beta_3)$ $= \boxed{0}$	$C_{44}-(\alpha_1+\beta_4)$
x_2	$C_{21}-(\alpha_2+\beta_1)$	$C_{22}-(\alpha_2+\beta_2)$	$C_{23}-(\alpha_2+\beta_3)$	$C_{24}-(\alpha_2+\beta_4)$ $= \boxed{0}$
x_3	$C_{31}-(\alpha_3+\beta_1)$	$C_{32}-(\alpha_3+\beta_2)$	$C_{33}-(\alpha_3+\beta_3)$	$C_{34}-(\alpha_3+\beta_4)$ $= \not{0}$
x_4	$C_{41}-(\alpha_4+\beta_1)$ $= \boxed{0}$	$C_{42}-(\alpha_4+\beta_2)$ $= \not{0}$	$C_{43}-(\alpha_4+\beta_3)$	$C_{44}-(\alpha_4+\beta_4)$

Scanned with OKEN Scanner

În matricea $\widetilde{A}$ am obținut zerourile pe care le-am încadrat sau tăiat cu ocazia determinării suportului minim.

A obține matricea $\widetilde{\widetilde{A}}$ revine la a suprima din matricea $\widetilde{A}$ liniile și coloanele suportului.

Elementul r reprezintă cel mai mic element netăiat din matricea $\widetilde{A}$.

Matricea $\widetilde{\widetilde{A}}$ am văzut că are următoarea formă:

$$\widetilde{\widetilde{A}} = \begin{array}{c|c|c|c|c} & y_1 & y_2 & y_3 & y_4 \\ \hline x_1 & C_{11} - (\alpha_1 + \beta_1) = \widetilde{C}_{11} & C_{12} - (\alpha_1 + \beta_2) = \widetilde{C}_{12} & C_{13} - (\alpha_1 + \beta_3) = \widetilde{C}_{13} & C_{14} - [\alpha_1 + (\beta_4 - r)] = \widetilde{C}_{14} + r \\ \hline x_2 & C_{21} - [(\alpha_2 + r) + \beta_1] = \widetilde{C}_{21} - r & C_{22} - [(\alpha_2 + r) + \beta_2] = \widetilde{C}_{22} - r & C_{23} - [(\alpha_2 + r) + \beta_3 = \widetilde{C}_{23} - r & C_{24} - [(\alpha_2 + r) + (\beta_2 - r)] = \widetilde{C}_{24} \\ \hline x_3 & C_{31} - [(\alpha_3 + r) + \beta_1] = \widetilde{C}_{31} - r & C_{32} - [(\alpha_3 + r) + \beta_2] = \widetilde{C}_{32} - r & C_{33} - [(\alpha_3 + r) + \beta_3] = \widetilde{C}_{23} - r & C_{34} - [(\alpha_3 + r) + (\beta_3 - r)] = \widetilde{C}_{34} \\ \hline x_4 & C_{41} - (\alpha_4 + \beta_1) = \widetilde{C}_{41} & C_{42} - (\alpha_4 + \beta_2) = \widetilde{C}_{42} & C_{43} - (\alpha_4 + \beta_3) = \widetilde{C}_{43} & C_{44} - [\alpha_3 + (\beta_3 - r)] = \widetilde{C}_{44} + r \end{array}$$

Analizînd elementele matricei $\widetilde{\widetilde{A}}$, deducem un mod simplu de calcul a acestora, pornind de la elementele corespunzătoare ale matricei $\widetilde{A}$ și anume:

a) Considerăm minorantul r al elementelor netăiate din matricea $\widetilde{A}$.

b) Adăugăm acest minorant elementelor duble tăiate și le scădem din elementele netăiate, iar elementele tăiate o singură dată rămîn neschimbate.

Cele demonstrate mai sus precum și observațiile făcute stau la baza algoritmului ungar (Egervahry-Kuhn) pentru determinarea cuplajului maxim într-un graf simplu corespunzător unui cost total minim.

Scanned with OKEN Scanner

§ 4. ALGORITMUL UNGAR PENTRU DETERMINAREA CUPLAJULUI MAXIM

Etapa I. Se scade minorantul fiecărei linii din toate elementele liniei respective, obținînd în acest fel cîte un zero situat pe fiecare linie a matricei A. În cazul în care, în urma acestei operații, există coloane care nu conțin zerouri, se scade minorantul fiecărei coloane din toate elementele respective.

Evident că în felul acesta s-a obținut cel puțin un zero situat pe fiecare linie și fiecare coloană din matricea A.

Etapa a II-a. Fie $\widetilde{A}$ matricea obținută din A, după parcurgerea etapei I. Determinăm un cuplaj maxim în $\widetilde{A}$, prin procedeul de încadrare și tăiere a zerourilor.

Dacă se obține cîte un zero încadrat pe fiecare linie și pe fiecare coloană, ceea ce este echivalent cu a spune că numărul elementelor suportului $\sigma(A)$ este egal cu dimensiunea matricei n, s-a obținut soluția optimă (cuplajul maxim).

În caz contrar, se trece la:

Etapa a III-a. Căutăm suportul minim printr-o operație de marcare a liniilor și coloanelor și anume:

— Marcăm liniile care nu conțin zerouri încadrate și coloanele care au zerouri tăiate pe liniile marcate.

— Marcăm liniile cu zero încadrat pe coloanele marcate. Operația de marcare continuă cît este posibil.

Etapa a IV-a. În matricea $\widetilde{A}$ tăiem toate liniile nemarcate și coloanele marcate.

Minorantul elementelor netăiate îl adunăm elementelor dublu tăiate, îl scădem din elementele netăiate, lăsînd neschimbate elementele simplu tăiate; obținem astfel matricea $\widetilde{\widetilde{A}}$.

În matricea $\widetilde{\widetilde{A}}$ repetăm operația de încadrare și tăiere a zerourilor. Dacă după această operație $\sigma(A) = n$, algoritmul ia sfîrșit.

În caz contrar, se reiau operațiile corespunzătoare etapelor a III-a și a IV-a.

Scanned with OKEN Scanner

Exemplul 1. Fie 6 muncitori calificați, ce urmează a fi repartizați pe 6 mașini. Cunoscînd că muncitorii pot lucra numai pe anumite mașini și că rezultatele muncii fiecăruia dintre ei variază în funcție de mașina pe care este repartizat, să se facă o astfel de repartizare, încît un muncitor să lucreze pe o singură mașină, iar productivitatea totală a muncii să fie maximă.

În tabela de mai jos sînt dați (pe bază de observații) indicii de productivitate în procente față de normă, pentru fiecare muncitor și fiecare mașină în parte.

$T_1 =$

	y_1	y_2	y_3	y_4	y_5	y_6
x_1	80	100	112	$-\infty$	95	107
x_2	103	$-\infty$	70	$-\infty$	65	112
x_3	90	65	83	104	72	$-\infty$
x_4	84	120	$-\infty$	110	100	80
x_5	$-\infty$	96	90	105	$-\infty$	100
x_6	90	85	$-\infty$	100	112	111

În tabelul T_1, în unele căsuțe a fost pus $(-\infty)$, acolo unde muncitorul nu este pregătit pentru a lucra pe mașina respectivă. Astfel, muncitorul (x_1) nu poate lucra pe mașina (y_4), și ca atare în căsuța (x_1, y_4) s-a trecut $(-\infty)$.

Deoarece problema cere determinarea repartiției corespunzătoare maximului de productivitate, vom înlocui tabelul T_1 cu matricea A, ale cărei elemente se obțin scăzînd fiecare element al tabelului T_1 din elementul maxim al acestuia.

Scanned with OKEN Scanner

$$A = \begin{array}{c|cccccc} & y_1 & y_2 & y_3 & y_4 & y_5 & y_6 \\ x_1 & 40 & 20 & 8 & \infty & 25 & 13 \\ x_2 & 17 & \infty & 50 & \infty & 55 & 8 \\ x & 30 & 44 & 37 & 16 & 48 & \infty \\ x_4 & 36 & 0 & \infty & 10 & 20 & 40 \\ x_5 & \infty & 24 & 30 & 15 & \infty & 20 \\ x_6 & 30 & 35 & \infty & 20 & 8 & 9 \end{array}$$

la matricea A aplicăm algoritmul Egervahry-Kuhn.

Etapa I. Obținerea de zerouri:

$$\begin{array}{c|cccccc} & y_1 & y_2 & y_3 & y_4 & y_5 & y_6 \\ x_1 & 32 & 12 & 0 & \infty & 17 & 5 \\ x_2 & 9 & \infty & 42 & \infty & 47 & 0 \\ x_3 & 14 & 39 & 21 & 0 & 32 & \infty \\ x_4 & 36 & 0 & \infty & 10 & 20 & 40 \\ x_5 & \infty & 9 & 15 & 0 & \infty & 5 \\ x_6 & 22 & 27 & \infty & 12 & 0 & 1 \end{array}$$

Întrucît nu s-au obținut zerouri și pe fiecare coloană, trebuie să scădem minoranții și de pe coloane și obținem matricea:

$$\widetilde{A} = \begin{array}{c|cccccc} & y_1 & y_2 & y_3 & y_4 & y_5 & y_6 \\ x_1 & 23 & 12 & 0 & \infty & 17 & 5 \\ x_2 & 0 & \infty & 42 & \infty & 47 & 0 \\ x_3 & 5 & 39 & 21 & 0 & 32 & \infty \\ x_4 & 27 & 0 & \infty & 10 & 20 & 40 \\ x_5 & \infty & 9 & 15 & 0 & \infty & 5 \\ x_6 & 13 & 27 & \infty & 12 & 0 & 1 \end{array}$$

Scanned with OKEN Scanner

Etapa a II-a. Trecem acum la operația de încadrare și tăiere a zerourilor:

$\widetilde{A} =$

	y_1	y_2	y_3	y_4	y_5	y_6	
x_1	~~23~~	~~12~~	~~$\boxed{0}$~~	~~∞~~	~~17~~	~~5~~	
x_2	~~0~~	~~∞~~	~~42~~	~~∞~~	~~47~~	~~$\boxed{0}$~~	
x_3	5	39	21	~~0~~	32	∞	*
x_4	~~27~~	~~$\boxed{0}$~~	~~∞~~	~~10~~	~~20~~	~~40~~	
x_5	∞	9	15	$\boxed{0}$	∞	5	*
x_6	~~13~~	~~27~~	~~∞~~	~~12~~	~~$\boxed{0}$~~	~~1~~	
				*			

În același tabel marcăm linia x_3, care nu conține nici un zero încadrat, coloana y_4, care conține zero tăiat pe linia x_3, apoi linia x_5, care conține zero încadrat pe coloana marcată y_4. Procedeul de marcare nu mai poate continua.

Suprimăm acum liniile nemarcate:

x_1, x_2, x_4, x_6 și coloana marcată y_4.

Etapa a III-a. Minorantul elementelor netăiate este 5.

Obținem matricea $\widetilde{\widetilde{A}}$

$\widetilde{\widetilde{A}} =$

	y_1	y_2	y_3	y_4	y_5	y_6
x_1	23	12	$\boxed{0}$	∞	17	5
x_2	$\boxed{0}$	∞	42	∞	47	~~0~~
x_3	~~0~~	34	16	$\boxed{0}$	27	∞
x_4	27	$\boxed{0}$	∞	15	20	40
x_5	∞	4	10	~~0~~	∞	$\boxed{0}$
x_6	13	27	∞	17	$\boxed{0}$	1

Repetăm operația de încadrare și tăiere a zerourilor din matricea $\widetilde{\widetilde{A}}$.

Deoarece am obținut cîte un zero încadrat pe fiecare linie și pe fiecare coloană, am obținut cuplajul maxim corespunzător productivității totale maxime, redat în graful din figura III.B.5.1.

Scanned with OKEN Scanner

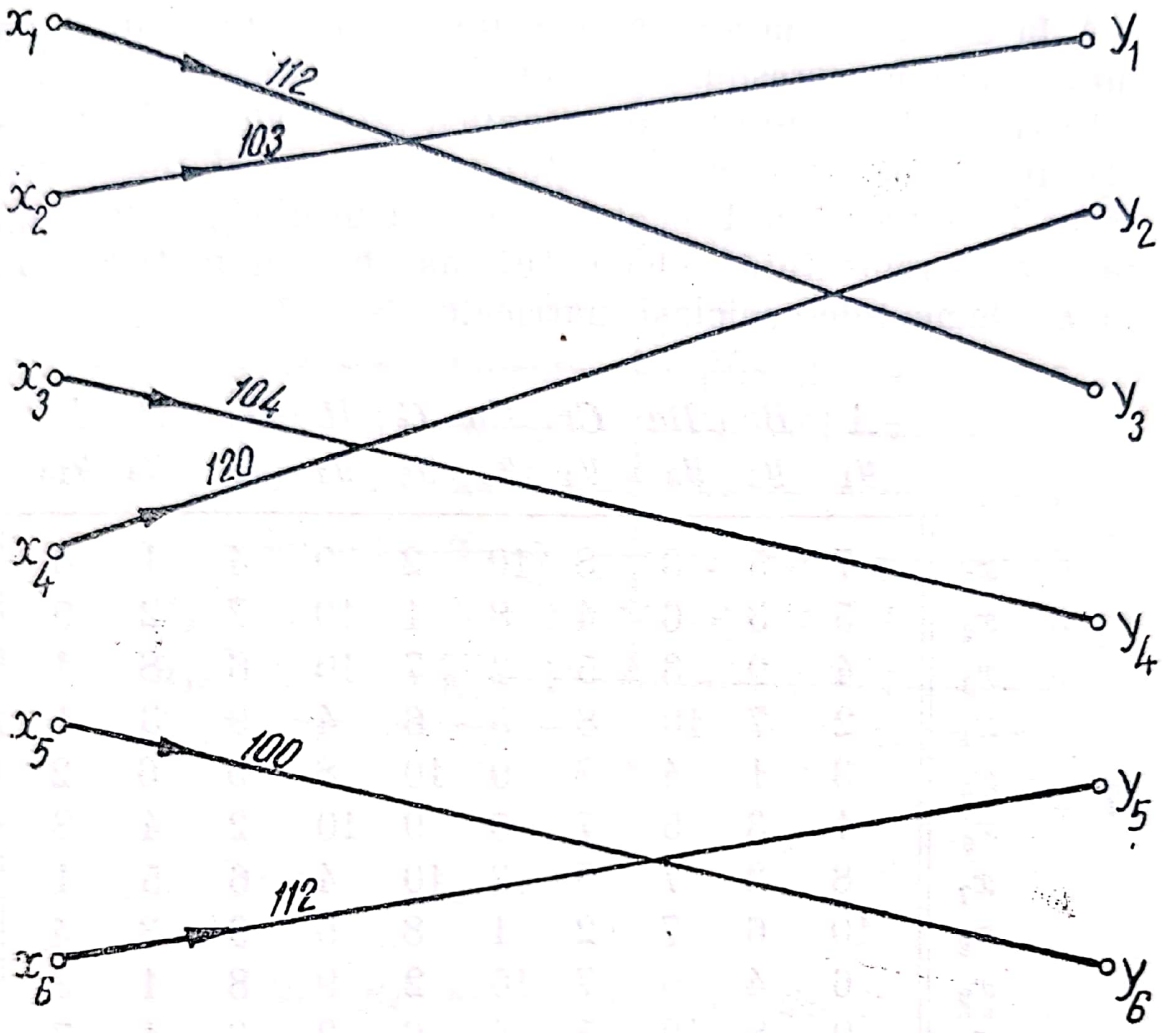


Fig. III.B.5.1

Exemplul 2. Un număr de zece absolvenți ai secției mecanizare, de pe lîngă facultatea de calcul economic și cibernetică economică, urmează a fi repartizați în zece centre industriale: Hunedoara, Arad, Galați, Reșița, Craiova, Tg. Mureș, Bacău, Constanța, Onești, Brașov.

În dorința de a se face o repartiție care să țină seamă și de preferințele candidaților, s-a dat fiecărui absolvent să-și noteze cu numere de la unu la zece ordinea preferințelor. Numerele cele mai mari reprezintă preferința cea mai mare. Pe baza acestor notări a fost întocmit tabelul T. Se cere să se întocmească o astfel de repartiție, încît să fie cît mai mult satisfăcute cerințele absolvenților.

În tabel candidații au fost notați cu x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, iar centrele unde urmează să fie repartizați cu y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}.

Scanned with OKEN Scanner

A face o asemenea repartiție înseamnă a determina un cuplaj maxim corespunzător unui „cost" total maxim.

Întrucît algoritmul ungar se aplică problemelor de repartiție în care se cere un cost total minim, transformăm matricea T în matricea A, pentru care elementul $C_{ij} = 10 - t_{ij}$, unde zece reprezintă elementul maxim din matricea T, iar t_{ij} elementele aceleiași matrice.

$T =$

	A y_1	Br y_2	Ba y_3	Cr y_4	Co y_5	G y_6	H y_7	O y_8	R y_9	T y_{10}
x_1	7	5	3	8	10	2	9	4	1	6
x_2	5	3	6	4	8	1	10	7	2	9
x_3	4	9	3	5	2	7	10	6	8	1
x_4	2	7	10	8	5	6	4	9	3	1
x_5	3	1	4	7	9	10	8	5	6	2
x_6	1	3	5	7	6	9	10	2	4	8
x_7	8	2	7	9	3	10	4	6	5	1
x_8	10	6	7	2	1	8	5	9	3	4
x_9	6	4	5	7	10	2	9	8	1	3
x_{10}	9	8	10	5	4	6	2	3	1	7

$A =$

	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}
x_1	3	5	7	2	0	8	1	6	9	4
x_2	5	7	4	6	2	9	0	3	8	1
x_3	6	1	7	5	8	3	0	4	2	9
x_4	8	3	0	2	5	4	6	1	7	9
x_5	7	9	6	3	1	0	2	5	4	8
x_6	9	7	5	3	4	1	0	8	6	2
x_7	2	8	3	1	7	0	6	4	5	9
x_8	0	4	3	8	9	2	5	1	7	6
x_9	4	5	5	3	0	8	1	2	9	7
x_{10}	1	2	0	5	6	4	8	7	9	3

Scanned with OKEN Scanner

$A =$

	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	
x_1	3	4	7	1	[0]	8	1	5	7	3	*
x_2	~~5~~	~~6~~	~~4~~	~~5~~	~~2~~	~~9~~	~~8~~	~~2~~	~~6~~	[0]	
x_3	~~6~~	[0]	~~7~~	~~4~~	~~8~~	~~3~~	~~0~~	~~3~~	~~0~~	~~8~~	
x_4	~~8~~	~~2~~	~~0~~	~~1~~	~~5~~	~~4~~	~~6~~	[0]	~~5~~	~~8~~	
x_5	~~7~~	~~8~~	~~6~~	~~2~~	~~1~~	[0]	~~2~~	~~4~~	~~2~~	~~7~~	
x_6	~~9~~	~~5~~	~~5~~	~~2~~	~~4~~	~~1~~	[0]	~~7~~	~~4~~	~~1~~	
x_7	~~2~~	~~7~~	~~3~~	[0]	~~7~~	~~0~~	~~6~~	~~3~~	~~3~~	~~8~~	
x_8	[0]	~~3~~	~~3~~	~~7~~	~~9~~	~~2~~	~~5~~	~~0~~	~~5~~	~~5~~	
x_9	4	5	5	2	~~0~~	8	1	1	7	6	*
x_{10}	~~1~~	~~1~~	[0]	~~4~~	~~6~~	~~4~~	~~8~~	~~7~~	~~7~~	~~2~~	
					*						

$\widetilde{\widetilde{A}} =$

	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	
x_1	2	3	6	~~0~~	[0]	7	~~0~~	1	6	2	*
x_2	~~5~~	~~6~~	~~4~~	~~5~~	~~3~~	~~9~~	~~8~~	~~2~~	~~6~~	[0]	
x_3	~~6~~	~~0~~	~~7~~	~~4~~	~~9~~	~~3~~	~~1~~	~~3~~	[0]	~~8~~	
x_4	8	2	~~0~~	1	6	4	6	[0]	5	8	*
x_5	7	8	6	2	2	[0]	2	4	2	7	*
x_6	9	6	5	2	5	~~1~~	[0]	7	4	1	*
x_7	2	7	3	[0]	8	~~0~~	6	3	3	8	*
x_8	[0]	~~3~~	~~3~~	~~7~~	~~10~~	~~2~~	~~5~~	~~0~~	~~5~~	~~5~~	
x_9	3	4	4	1	~~0~~	7	~~0~~	~~0~~	6	5	*
x_{10}	1	1	[0]	4	7	4	8	7	7	2	*
			*	*	*	*	*	*			

Scanned with OKEN Scanner

$\overset{\approx}{A} =$

	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}
x_1	1	2	6	~~0~~	[0]	7	~~0~~	4	5	1
x_2	5	6	5	6	4	10	1	3	6	[0]
x_3	6	~~0~~	8	5	10	4	1	4	[0]	8
x_4	7	1	[0]	1	6	4	6	~~0~~	4	7
x_5	6	7	6	2	2	[0]	2	4	1	6
x_6	8	5	5	2	5	1	[0]	7	3	~~0~~
x_7	1	6	3	[0]	8	~~0~~	6	3	2	7
x_8	[0]	3	4	8	11	3	6	1	5	5
x_9	2	3	4	1	~~0~~	7	~~0~~	[0]	5	4
x_{10}	~~0~~	[0]	~~0~~	4	7	4	8	7	6	1

Matricea $\overset{\approx}{A}$ conține soluția optimă redată în graful din figura III.B.5.2.

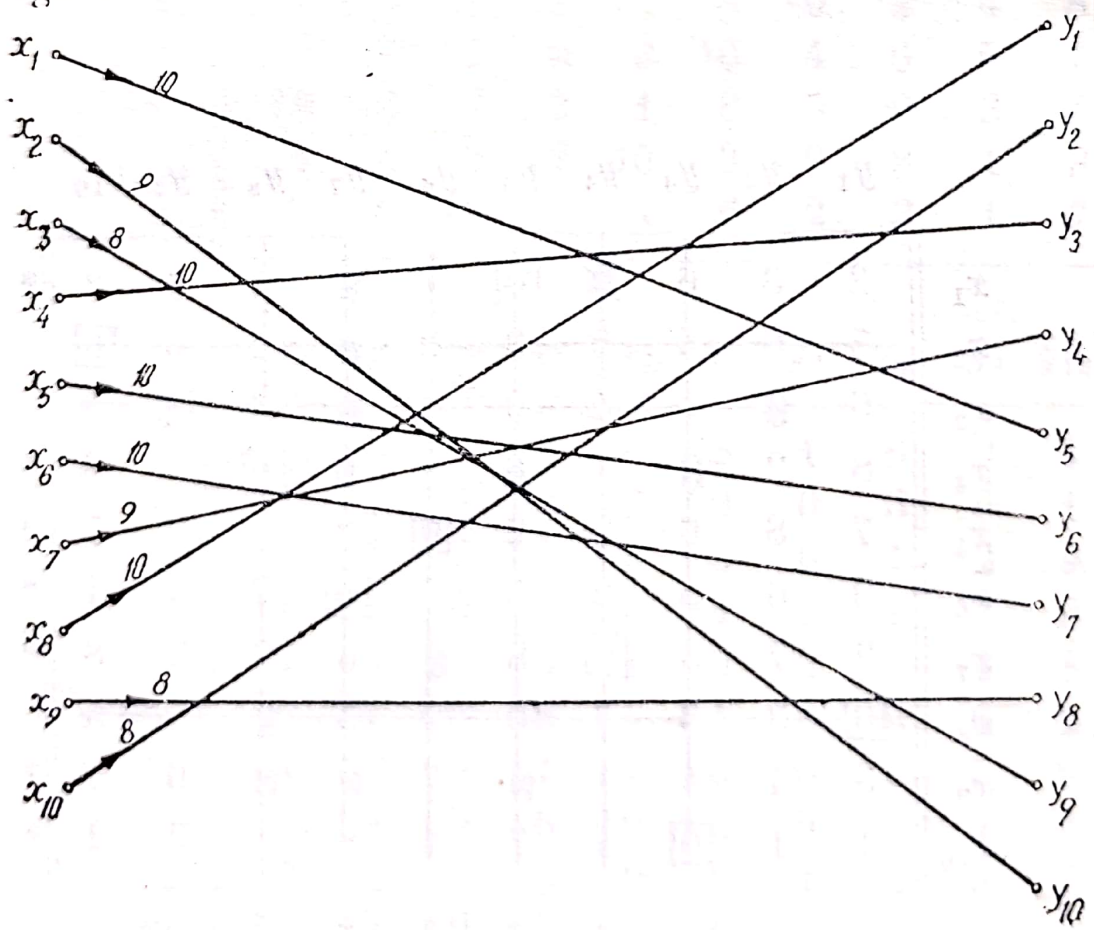


Fig. III.B.5.2

Exemplul 3. Vom trata acum un model care diferă de cele precedente prin modul de prelucrare a datelor, în scopul aplicării aceluiași algoritm.

Se cunoaște cît de importantă este problema traficului feroviar și că aceasta este strîns legată de personalul afectat fiecărei garnituri de tren (mecanici, fochiști, taxatori, personal de îngrijire, personal poștă etc.). Eforturile pe care aceștia le depun pentru buna deservire sînt legate nu numai de efortul fizic, și de consumul nervos, ci și de cel de ordin social, în sensul că aceștia trebuie să-și părăsească familiile pe anumite durate de timp. Grija pentru o raționalizare a timpului de lucru, cît și pentru despărțirea de familiile lor, este o preocupare permanentă din partea organizatorilor acestui tip de transport.

În această ordine de idei, să considerăm personalul care deservește trenurile care parcurg rețelele București-Timișoara.

Luăm timpul necesar parcurgerii distanței București-Timișoara egal cu 9 ore. Practica însă impune ca după o cursă efectuată, personalul să-și reia activitatea după cel puțin 6 ore de odihnă.

Plecările din stația București și sosirile în stația Timișoara, și invers, sînt redate în tabelul de mai jos:

Plecarea din București	Trenul	Sosirea în Timișoara
0^{05}	x_1	14
06^{30}	x_2	15^{30}
09^{15}	x_3	18^{15}
11^{45}	x_4	20^{45}
17	x_5	0^{02}

Sosirea în București	Trenul	Plecarea din Timișoara
14	y_1	0^{05}
17	y_2	0^{08}
23	y_3	14
0^{03}	y_4	18
08^{30}	y_5	23^{30}

Scanned with OKEN Scanner

Se știe că pe această rută, trenurile sînt deservite de personal cu domiciliul fie în București, fie în Timișoara, și că intervalul de timp petrecut cu trenul ce-l deservește este același, de 9 ore.

Problemele ce se pun în legătură cu buna organizare a timpului personalului de deservire sînt legate de următoarele considerente:

a) Timpul petrecut în afara domiciliului — abstracție făcînd de cele 9 ore de serviciu — să fie cît mai mic.

b) După parcurgerea unei curse, personalul trebuie să aibă un timp minim de odihnă de cel puțin 6 ore pînă la reluarea unei noi curse.

c) Personalul să nu rămînă în repaus mai mult de 24 de ore.

Pentru o asemenea situație, se pune problema stabilirii domiciliului personalului de deservire la liniile pe care acesta trebuie să le deservească, astfel ca timpul total al întregului personal de deservire a rutei București-Timișoara și retur, format din timpul de așteptare la întoarcere, să fie minim, știind că acest timp trebuie să varieze între 6 și 24 de ore.

Pentru a da răspunsul problemei puse întocmim două tabele, unul în ipoteza că întregul personal de deservire are domiciliul în București, și celălalt, dacă același personal are domiciliul în Timișoara.

Obținem astfel tabelele T_1 și T_2.

Personalul cu domiciliul în București

$T_1 =$

	y_1	y_2	y_3	y_4	y_5
x_1	15	18	0	4	9,5
x_2	8,5	11,5	17,5	21,5	3
x_3	10,75	13,75	19,75	23,75	5,25
x_4	8,25	11,25	17,25	21,25	2,75
x_5	3	5	12	16	21,50

Personalul cu domiciliul în Timișoara

$T_2 =$

	y_1	y_2	y_3	y_4	y_5
x_1	15	16,5	19,25	21,75	3
x_2	12	13,5	16,25	18,75	0
x_3	6	7,5	10,25	12,75	18
x_4	2	3,5	6,25	8,75	14
x_5	20,5	22	0,75	3,15	8,5

Observăm că personalul care locuiește în București și deservește trenul x_3 la dus și trenul y_2 la întors, rămîne în Timișoara 13,75 ore; sau în tabelul T_2, personalul care locuiește în Timișoara și deservește, la dus, trenul y_4 și, la întors, trenul x_3 rămîne în București 12,75 ore.

Vom construi un nou tabel T_3, în care, în căsuța (i,j), vom lua timpul cel mai mic corespunzător căsuțelor (i,j) din prima și a doua, cu condiția ca să fie cuprins între 6 și 24 de ore.

$T_3 =$

	y_1	y_2	y_3	y_4	y_5
x_1	15	16,5	19,25	21,75	9,5
x_2	8,5	11,5	16,25	18,75	24
x_3	6	7,5	10,25	12,75	18
x_4	8,25	11,25	6,25	8,75	14
x_5	20,5	22	12	16	8,5

În căsuța (x_2, y_5) din tabelul T_3 am trecut timpul de 24 de ore, deoarece, comparînd valorile corespunzătoare aceleiași poziții din tabelele T_1 și T_2, avem de ales între 3 ore și zero ore. Or, nici prima, nici a doua valoare nu respectă condiția de a fi mai mare ca 6 ore.

Ținînd seama însă că zeroul din această căsuță corespunde personalului de tren care locuiește în Timișoara, deservește trenul y_5 pînă în București, ajunge aici la ora 17, iar trenul x_5 se întoarce tot la ora 17, rezultă că acest personal așteaptă în București fie zero ore — ceea ce nu respectă condițiile impuse —, fie 24 de ore, timp pe care-l acceptăm la limită.

Scanned with OKEN Scanner

Din acest moment, rezolvarea problemei propuse se face utilizînd algoritmul Egervahry-Kuhn, cu privire la cuplajul maxim în rețea.

Etapa I. Scădem minorantul fiecărei linii și apoi minorantul fiecărei coloane din linia, respectiv coloana considerată.

Obținem tabelul T_4, apoi T_5.

$T_4 =$

	y_1	y_2	y_3	y_4	y_5
x_1	5,5	7	9,75	12,25	0
x_2	0	3	7,75	10,25	15,5
x_3	0	1,5	4,25	6,75	12
x_4	2	5	0	2,5	7,75
x_5	12	13,5	3,5	7,5	0

$T_5 =$

	y_1	y_2	y_3	y_4	y_5
x_1	5,5	5,5	9,75	9,75	0
x_2	0	1,5	7,75	7,75	15,5
x_3	0	0,5	4,25	4,25	12
x_4	2	3,5	0	0	7,75
x_5	12	12	3,5	5	0

Etapa a II-a. Operația de încadrare și tăiere a zerourilor și operația de marcare a liniilor și coloanelor.

Obținem tabelul T_6

$T_6 =$

	y_1	y_2	y_3	y_4	y_5	
x_1	5,5	5,5	9,75	9,75	~~0~~	*
x_2	~~[0]~~	~~1,5~~	~~7,75~~	~~7,75~~	~~15,5~~	
x_3	~~0~~	~~[0]~~	~~4,25~~	~~4,25~~	~~12~~	
x_4	~~2~~	~~3,5~~	~~[0]~~	~~0~~	~~7,75~~	
x_5	12	12	3,5	5	[0]	*
					*	

În tabelul T_6 am suprimat liniile nemarcate și coloanele marcate, adică liniile x_2, x_3 și x_4 și coloana y_5.

Etapa a III-a. Minorantul elementelor netăiate, adică elementul 3,5 de pe linia x_5, coloana y_3, îl scădem din elementele netăiate, îl adăugăm elementelor dublu tăiate, iar elementele tăiate o singură dată le lăsăm neschimbate. În felul acesta, obținem tabelul T_7.

T_7

	y_1	y_2	y_3	y_4	y_5
x_1	5,5	5,5	9,75	9,75	0
x_2	0	1,5	7,75	7,75	19
x_3	0	0	4,25	4,25	15,5
x_4	2	3,5	0	0	11,25
x_5	8,5	8,5	0	1,5	0

În tabelul T_7 repetăm operațiile din etapa I, obținînd tabelul T_8.

T_8

	y_1	y_2	y_3	y_4	y_5
x_1	5,5	5,5	9,75	9,75	$\boxed{0}$
x_2	$\boxed{0}$	1,5	7,75	7,75	19
x_3	~~0~~	$\boxed{0}$	4,25	4,25	15,5
x_4	2	3,5	~~0~~	$\boxed{0}$	11,25
x_5	8,5	8,5	$\boxed{0}$	1,5	~~0~~

În tabelul T_8 am obținut cîte un zero încadrat pe fiecare linie și pe fiecare coloană; ca atare, s-a obținut soluția optimă. Repartiția este următoarea:

Domiciliul	Trenuri	Timp de așteptare
București	y_1	8,5 ore în Timișoara
Timișoara	y_2	7,5 ore în București
București	y_3	12,0 ore în Timișoara
Timișoara	y_4	8,75 ore în București
București	y_5	9,5 ore în Timișoara

Scanned with OKEN Scanner

Soluţia optimă sugerează ideea angajării personalului în funcţie de domiciliu şi de linia pe care o va deservi. Astfel, linia y_3 va fi deservită de un personal cu domiciliul în Bucureşti şi va aştepta în Timişoara, pînă la reluarea cursei către Bucureşti, numai 12 ore.

Exemplul 4. Să presupunem că într-un anumit sezon în sectorul viticol sînt prevăzute lucrări diferite. În afara retribuirii în natură a lucrătorilor, aceştia sînt retribuiţi şi în bani. Costurile unei ore-muncă, care diferă de la o lucrare la alta, sînt date în tabelul 1. Se cere să se facă repartizarea a 5 lucrători pe aceste lucrări, astfel încît cîştigul total al acestora să fie maxim, ştiind că fiecare lucrare trebuie executată de un muncitor, şi numai de unul.

$T_1 =$

	y_1	y_2	y_3	y_4	y_5
x_1	9	8	6	5	10
x_2	2	6	10	7	9
x_3	3	1	4	9	10
x_4	8	1	5	8	9
x_5	10	9	7	8	2

Pentru aplicarea algoritmului, vom scădea elementele tabelului T_1 din majorantul acestui tabel, obţinînd tabelul T_1^*

$T_1^* =$

	y_1	y_2	y_3	y_4	y_5
x_1	1	2	4	5	0
x_2	8	4	0	3	1
x_3	7	9	6	1	0
x_4	2	9	5	2	1
x_5	2	1	3	2	8

Pentru a determina repartiţia optimă, aplicăm algoritmul ungar tabelului T_1^*.

Scanned with OKEN Scanner

Etapa 2

$T_2 =$

	y_1	y_2	y_3	y_4	y_5
x_1	1	1	4	4	0
x_2	8	3	0	2	1
x_3	7	8	6	0	0
x_4	2	8	5	1	1
x_5	0	0	3	1	8

$T_3 =$

	y_1	y_2	y_3	y_4	y_5
x_1	1	1	4	4	0
x_2	8	3	0	2	1
x_3	7	8	6	0	0
x_4	1	7	4	0	0
x_5	0	0	3	1	8

$T_4 =$

	y_1	y_2	y_3	y_4	y_5	
x_1	1	1	4	4	[0]	*
x_2	~~8~~	~~3~~	~~0~~	~~2~~	~~1~~	
x_3	7	8	6	[0]	~~0~~	*
x_4	1	7	4	~~0~~	~~0~~	*
x_5	~~[0]~~	[0]	3	~~1~~	~~8~~	
				*	*	

$T_5 =$

	y_1	y_2	y_3	y_4	y_5
x_1	~~0~~	~~0~~	3	4	[0]
x_2	8	3	[0]	3	2
x_3	6	7	5	[0]	~~0~~
x_4	[0]	6	3	~~0~~	~~0~~
x_5	~~0~~	[0]	3	2	9

Scanned with OKEN Scanner

În tabelul T_5 am obținut cîte un singur zero încadrat pe fiecare linie și pe fiecare coloană, adică soluția optimă, pe care o redăm în graful din figura B.5.3.

Reluînd tabelul T_5 și procedînd la o nouă alegere a zerourilor încadrate și tăiate, obținem o altă soluție optimă, dată de tabelul T_6.

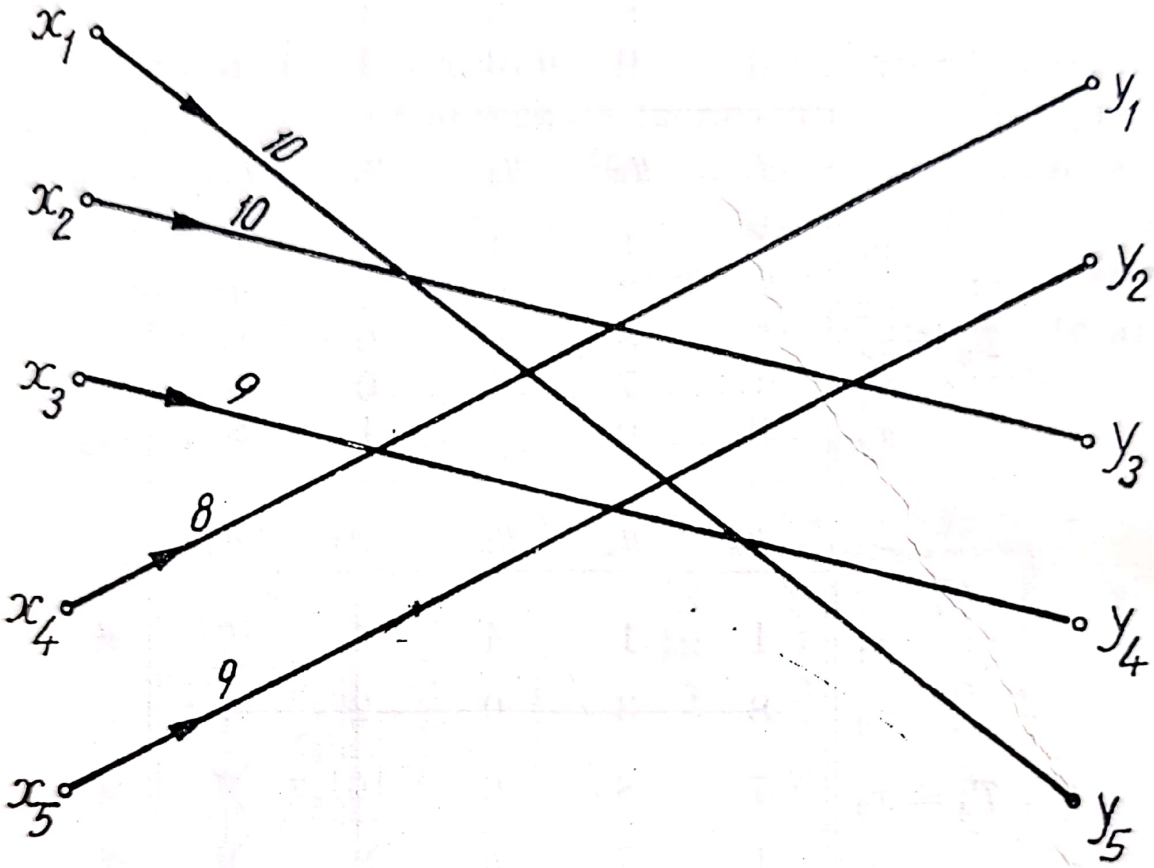


Fig. III.B.5.3

$T_6 =$

	y_1	y_2	y_3	y_4	y_5
x_1	~~0~~	[0]	3	4	~~0~~
x_2	8	3	[0]	3	2
x_3	6	7	5	~~0~~	[0]
x_4	~~0~~	6	3	[0]	~~0~~
x_5	[0]	~~0~~	3	2	9

§ 6. APLICAREA ALGORITMULUI UNGAR PENTRU REZOLVAREA PROBLEMELOR DE TRANSPORT

Pînă acum ne-am ocupat de aplicarea algoritmului ungar pentru probleme de repartiție, probleme în care matricea costurilor C_{ij} $(i,j = 1, \ldots, n)$ era de formă pătrată.

În cele ce urmează vom arăta cum se folosește algoritmul ungar în rezolvarea unor probleme tot de repartiție, în cazul în care matricea costurilor C_{ij} este de formă dreptunghiulară $(i = 1, \ldots, n;\quad j = 1, \ldots, n)$.

Este cunoscut că asemenea matrice sînt frecvent asociate problemelor de tip transport.

În general, într-o problemă de transport se dau m centre expediție A_i și n centre de consum B_j.

Se cunoaște disponibilul a_i corespunzător fiecărui centru A_i și necesarul b_j corespunzător fiecărui centru de consum B_j. De asemenea, se cunosc costurile C_{ij} ale transportului unității de produs de la centrul A_i la centrul B_j, sau distanțele.

În această situație, se cere să se întocmească un plan de repartiție, astfel încît să fie transportat tot disponibilul și să fie satisfăcut tot necesarul iar costul total al transportului să fie minim.

Evident că pentru a fi satisfăcute centrele B_j trebuie să avem realizată condiția

$$\sum_{i=1}^{n} a_i \geqslant \sum_{j=1}^{n} b_j .$$

În calcule, se transformă inegalitatea în egalitate prin introducerea unor centre fictive de consum. Pe viitor vom considera totdeauna problema așa-zisă echilibrată adică:

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j = T.$$

Algoritmul constă în efectuarea unui număr de operații, care să conducă la obținerea unui număr de zerouri cel puțin egal cu $m \cdot n - (m + n - 1)$, iar soluția găsită să fie optimă $(x_{ij}^{*} \geqslant 0;\ f_{min.})$.

Scanned with OKEN Scanner

În matricea T_1 a costurilor se procedează la obținerea de zerouri pe fiecare linie și pe fiecare coloană, prin scăderea minoranților liniilor (coloanelor). În felul acesta vom obține o matrice T_2, care are cel puțin cîte un element egal cu zero în fiecare linie și în fiecare coloană. Dacă în matricea T_2 facem o repartiție folosind căsuțele cu costuri nule, vom obține o soluție inițială de bază. Soluția obținută s-ar putea să fie cea optimă sau nu. Pentru a stabili în ce situație ne aflăm, trecem la așa-zisa operație de acoperire a elementelor egale cu zero din matricea T_2, prin tăierea unui număr minim de linii și de coloane în felul următor:

a) Se taie pe rînd fiecare linie și coloanele ce mai conțin zerouri, cu condiția ca aceste zerouri să nu fie situate pe linia tăiată.

b) Se taie apoi grupe de cîte două linii și coloanele ce conțin zerouri nesituate pe cele două linii.

c) Se taie grupe de cîte 3 linii și coloanele ce conțin zerouri nesituate pe liniile tăiate.

d) Se taie apoi grupe de cîte 4, 5 etc. linii și coloanele care conțin zerouri în afara liniilor tăiate.

Este clar că dacă matricea asociată problemei de transport are dimensiunea $(m \times n)$ și ținînd seama de modul în care s-au făcut tăieturile, vom avea de tăiat un număr

$$N = C_m^1 + C_m^2 + \ldots + C_m^m = 2^m - 1$$

de posibilități. Pentru fiecare suprimare în parte se calculează cîte o sumă S_k $(k = 1, 2, \ldots, 2^m - 1)$, de forma

$$S_k = \sum_i a_i + \sum_j b_i,$$

în care i reprezintă liniile și j coloanele corespunzătoare, care au fost tăiate la acoperirea zerourilor pentru fiecare valoare a lui k.

După calculul acestora sume S_k , se pot ivi trei situații:

1) $S_i > T$,

2) $S_i = T$,

3) $S_i < T$.

Scanned with OKEN Scanner

Situația 1) nu poate fi luată în considerație, deoarece transportul nu trebuie să depășească necesarul.

Cazul 3) ne obligă să trecem la o îmbunătățire a repartiției, prin obținerea de noi zerouri.

Repartiția ia sfîrșit cînd obținem $S_k \geqslant T$.

Dacă avem una sau mai multe sume $S_k < T$, este evident că se va lua în considerație suma cea mai mică, deoarece ea oferă posibilitatea cea mai mare de transport pentru satisfacerea necesarului T și pentru epuizarea disponibilului T.

În acest scop reluăm matricea T_2, în care se taie efectiv liniile și coloanele corespunzătoare sumei $\min \{S_k \mid S_k < T\}$, obținîndu-se matricea T_3.

În matricea T_3 considerăm minorantul elementelor netăiate, pe care îl scădem din toate elementele rămase netăiate; adunăm elementele dublu tăiate (adică tăiate de o linie și coloană), iar elementele tăiate o singură dată vor rămîne neschimbate, obținîndu-se în acest fel matricea T_4.

Cu matricea T_4 se repetă aceleași operații ca și cu matricea T_2, analizîndu-se din nou sumele $S_k^{(1)}$ în comparație cu totalul T.

Operațiile se repetă, pînă cînd suma minimă $S_k^{(p)}$ este egală cu totalul T, adică:

$$\min \left\{ S_k^{(p)} \right\} = T = \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j.$$

Să aplicăm algoritmul ungar, astfel adaptat problemei de transport cu date din sectorul industrial poligrafic.

În cursul trimestrului IV al anului 1966, fabricile producătoare de hîrtie, precum și tipografiile beneficiare din cadrul industriei poligrafice au fost următoarele:

A_1 — Fabrica de hîrtie Bușteni,
A_2 — Combinatul de celuloză și hîrtie Brăila,
A_3 — Combinatul de celuloză și hîrtie Călărași,
A_4 — Fabrica de hîrtie și cartoane Piatra Neamț,
A_5 — Fabrica de hîrtie Petrești,
A_6 — Fabrica de hîrtie și celuloză Letea—Bacău,
A_7 — Combinatul de celuloză și hîrtie Palas-Constanța,
A_8 — Fabrica de hîrtie cretată Ghimbav-Brașov.

Scanned with OKEN Scanner

Aceste fabrici au aprovizionat cu hîrtie de scris și tipărit întreprinderile poligrafice din localitățile:

B_1 — București,
B_2 — Timișoara,
B_3 — Cluj,
B_4 — Oradea,
B_5 — Sibiu.

Cantitățile disponibile de hîrtie a_i, cantitățile necesare b_j, precum și distanțele C_{ij} în km, dintre centrele de expediție A_i ($i = 1, \ldots, 8$) și centrele de destinație B_j ($j = 1, \ldots, 5$) sînt date în tabelul T_0. Se vede că

$$\sum_{i=1}^{8} a_i = \sum_{j=1}^{5} b_j = 3347{,}2.$$

Problema ce ni se pune este de a întocmi un plan de repartiție, astfel încît întreaga activitate de transport exprimată în t/km să fie minimă.

Notînd cu x_{ij} cantitatea de hîrtie transportată de la fabrica A_i la tipografia B_j, obținem sistemul de condiții (1):

$$(1)\quad \begin{cases} x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = 756{,}9, \\ x_{21} + x_{22} + x_{23} + x_{24} + x_{25} = 610{,}1 \\ x_{31} + x_{32} + x_{33} + x_{34} + x_{35} = 734{,}5, \\ x_{41} + x_{42} + x_{43} + x_{44} + x_{45} = 912{,}8, \\ x_{51} + x_{52} + x_{53} + x_{54} + x_{55} = 116{,}7, \\ x_{61} + x_{62} + x_{63} + x_{64} + x_{65} = 113{,}7, \\ x_{71} + x_{72} + x_{73} + x_{74} + x_{75} = 44{,}3, \\ x_{81} + x_{82} + x_{83} + x_{84} + x_{85} = 58{,}2, \\ x_{11} + x_{21} + x_{31} + x_{41} + x_{51} + x_{61} + x_{71} + x_{81} = 2\,543, \\ x_{12} + x_{22} + x_{32} + x_{42} + x_{52} + x_{62} + x_{72} + x_{82} = 88{,}5, \\ x_{13} + x_{23} + x_{33} + x_{43} + x_{53} + x_{63} + x_{73} + x_{83} = 337{,}5, \\ x_{14} + x_{24} + x_{34} + x_{44} + x_{54} + x_{64} + x_{74} + x_{84} = 294, \\ x_{15} + x_{25} + x_{35} + x_{45} + x_{55} + x_{65} + x_{75} + x_{85} = 84{,}2. \end{cases}$$

Obiectivul problemei noastre este ca totalul t/km afectați repartiției să fie minim. Acest lucru, din punct de vedere matematic, revine la a minimiza funcția liniară:

Scanned with OKEN Scanner

$$f = 129\,x_{11} + 527\,x_{12} + 368\,x_{13} + 521\,x_{14} + 186\,x_{15} +$$
$$+ 199\,x_{21} + 733\,x_{22} + 696\,x_{23} + 849\,x_{24} + 455\,x_{25} +$$
$$+ 136\,x_{31} + 670\,x_{32} + 633\,x_{33} + 786\,x_{34} + 451\,x_{35} +$$
$$+ 362\,x_{41} + 863\,x_{42} + 530\,x_{43} + 685\,x_{44} + 511\,x_{45} +$$
$$+ 302\,x_{51} + 803\,x_{52} + 470\,x_{53} + 625\,x_{54} + 451\,x_{55} +$$
$$+ 450\,x_{61} + 290\,x_{62} + 150\,x_{63} + 303\,x_{64} + 110\,x_{65} +$$
$$+ 225\,x_{71} + 759\,x_{72} + 722\,x_{73} + 875\,x_{74} + 540\,x_{75} +$$
$$+ 174\,x_{81} + 515\,x_{82} + 339\,x_{83} + 476\,x_{84} + 157\,x_{85}\,,$$

respectîndu-se condițiile date de sistemul (1).

Pentru rezolvarea problemei considerăm tabelul T_1^1 format cu disponibilul, necesarul și distanțele de la centrele A_i la centrele B_j:

T_0

A_i \ B_j	B_1	B_2	B_3	B_4	B_5	Disponibil
A_1	129	527	368	521	186	756,9
A_2	199	733	696	849	455	610,1
A_3	136	670	633	786	451	734,5
A_4	362	863	530	685	511	912,8
A_5	302	803	470	625	451	116,7
A_6	450	290	150	303	110	113,7
A_7	225	759	722	875	540	44,3
A_8	174	515	339	476	157	58,2
Necesar	2 543	88,5	337,5	294	84,2	3 347,2

Aplicînd algoritmul Egervahry-Kuhn, scădem din elementele fiecărei linii ale matricei din tabelul T_0 minorantul corespunzător.

Scanned with OKEN Scanner

Obținem astfel tabelul T_1:

T_1

	B_1	B_2	B_3	B_4	B_5
A_1	0	398	239	392	57
A_2	0	534	497	650	256
A_3	0	534	497	650	315
A_4	0	501	168	323	149
A_5	0	501	168	323	149
A_6	340	180	40	193	0
A_7	0	534	497	650	315
A_8	17	358	182	319	0

Deoarece prin această operație coloanele 2, 3, 4 nu conțin zerouri scădem din elementele fiecărei coloane minorantul corespunzător.

Evident, coloanele 1 și 5 rămîn neschimbate, întrucît minoranții respectivi sînt nuli.

Tabelul 2, care rezultă, conține cel puțin un zero pe fiecare linie, respectiv coloană, și el conține soluția de pornire conform celor arătate:

T_2

	B_1	B_2	B_3	B_4	B_5
A_1	0	218	199	199	57
A_2	0	354	457	457	256
A_3	0	354	457	457	315
A_4	0	321	128	130	149
A_5	0	321	129	130	149
A_6	340	0	0	0	0
A_7	0	354	457	457	315
A_8	17	178	142	226	0

Scanned with OKEN Scanner

În tabelul T_2, acoperind linia i ($i = 1, 2, 3, 4, 5, 7,8$) și coloanele care conțin zerouri nesituate pe linia i, adică $j = 1, 2, 3, 4, 5$, obținem sumele:

$$S_i = \sum_{j=1}^{5} b_j + a_i > T (i = 1, 2, 3, 4, 5, 7, 8),$$

unde T este totalul disponibilului, respectiv necesarului (3 347,2). Acoperind linia a 6-a, deci și coloanele 1 și 5, obținem

$$S_6 = b_1 + b_5 + a_6 = 2\,740{,}9 < T.$$

Acoperind grupe distincte de cîte 2 linii și coloanele care conțin zerouri nesituate pe liniile acoperite, obținem

$$S_{ij} = a_i + a_j + \sum_{k=1}^{5} b_k > T,$$

oricare ar fi $i \neq j$.

Acoperind liniile 6 și 8 și deci și coloana 1, obținem

$$S_{6,8} = a_6 + a_8 + b_1 = 2\,714{,}9 < T.$$

Acoperind grupe distincte de cîte 3 linii și coloanele care conțin zerouri în afara liniilor acoperite, obținem:

$$S_{ijk} > T,$$

oricare ar fi $i \neq j \neq k$ ($i, j, k = 1, 2, 3, 4, 5, 7, 8$).

Pentru liniile 6, 7, 8, deci și coloana 1, obținem

$$S_{6,7,8} = a_6 + a_7 + a_8 + b_1 = 2\,759{,}2 < T.$$

Dintre sumele mai mici ca $T = 3\,347{,}2$, cea mai mică se obține prin suprimarea liniilor 6 și 8 și coloana 1:

$$S_{6,8} = 2\,714{,}9.$$

Conform algoritmului, în tabelul T_2 vom tăia liniile 6 și 8 și coloana 1. Minorantul elementelor netăiate (57) îl vom scădea din elementele netăiate și-l vom aduna elementelor dublu tăiate. Elementele tăiate o singură dată rămîn neschimbate. Rezultatele acestor operații sînt consemnate în tabelul T_3.

Scanned with OKEN Scanner

T_3

	B_1	B_2	B_3	B_4	B_5
A_1	0	161	142	142	0
A_2	0	297	400	400	199
A_3	0	297	400	400	258
A_4	0	264	71	73	92
A_5	0	264	71	73	92
A_6	397	0	0	0	0
A_7	0	297	400	400	258
A_8	74	178	142	126	0

Procedînd cu tabelul T_3 la fel cum am procedat cu tabelul T_2, adică acoperind cîte o linie, cîte 2, cîte 3 etc., și coloanele care conțin zerouri în afara liniilor suprimate, și făcînd sumele corespunzătoare, constatăm că cea mai mică sumă inferioară lui $T = 3\,347{,}2$ o obținem în cazul acoperirii liniei 6 și coloanelor 1 și 5:

$$S_6 = a_6 + b_1 + b_5 = 2\,740{,}9 < T.$$

Acoperind linia 6 și coloanele 1 și 5, și scăzînd minorantul elementelor neacoperite din aceste elemente, adunîndu-l la elementele dublu tăiate și lăsînd neschimbate elementele tăiate o singură dată, obținem tabelul:

T_4

	B_1	B_2	B_3	B_4	B_5
A_1	0	90	71	71	0
A_2	0	226	329	329	199
A_3	0	226	329	329	258
A_4	0	193	0	2	92
A_5	0	193	0	2	92
A_6	468	0	0	0	71
A_7	0	226	329	329	258
A_8	145	107	71	55	0

Scanned with OKEN Scanner

În tabelul T_4, făcînd calculele corespunzătoare, ajungem la $S_k < T$. Repetînd operațiile corespunzătoare acoperirii liniei 6 și coloanelor 1, 3, 5, obținem tabelul T_5:

T_5

	B_1	B_2	B_3	B_4	B_5
A_1	0	88	71	69	0
A_2	0	224	329	327	199
A_3	0	224	329	327	258
A_4	0	191	0	0	92
A_5	0	191	0	0	92
A_6	470	0	2	0	73
A_7	0	224	329	327	258
A_8	145	105	71	53	0

În tabelul T_5, toate sumele calculate ca mai sus sînt mai mari sau egale cu $T = 3347{,}2$; deci problema este rezolvată conform algoritmului ungar.

Soluțiile sînt multiple, deoarece s-au obținut mai mult de 12 zerouri.

În vederea construirii soluțiilor optime multiple, se trece la determinarea uneia dintre ele, făcîndu-se repartiția în condițiile respectării disponibilului și necesarului $\left(\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j = T\right)$. Apoi se modifică această soluție, folosind pe rînd cîte unul din zerourile rămase în afara primei soluții. În modul acesta, se obțin soluțiile optime, în cazul problemei.

Problema admite o dublă infinitate de soluții optime, deoarece, după cum se știe, o combinație liniară convexă a primelor n soluții ne dă tot o soluție optimă:

$$x = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3,$$

$$\lambda_i \geqslant 0, \text{ cu } \sum_{i=1}^{3} \lambda_i = 1,$$

x_1, x_2, x_3 fiind soluțiile optime.

Scanned with OKEN Scanner

Tabelul T_6 conține prima soluție optimă:

T_6

	B_1	B_2	B_3	B_4	B_5	D.
A_1	730,9				26	756,9
A_2	610,1					610,1
A_3	734,5					734,5
A_4	423,2		337,5	152,1		912,8
A_5				116,7		116,7
A_6		88,5		25,2		113,7
A_7	44,3					44,3
A_8					58,2	58,2
N	2 343,0	88,5	337,5	294	84,2	3 347,2

Total t/km = 882 028,9

Tabelul T_7 conține a 2-a soluție optimă:

T_7

	B_1	B_2	B^3	B_4	B^5	D.
A_1	730,9				26	756,9
A_2	610,1					610,1
A_3	734,5					734,5
A_4	423,2		220,8	268,8		912,8
A_5			116,7			116,7
A_6		88,5		25,2		113,7
A_7	44,3					44,3
A_8					58,2	58,2
N	2 543	88,5	337,5	294	84,2	3 347,2

Total t/km = 882 028,9.

Tabelul T_8 conține cea de-a 3-a soluție optimă:

Scanned with OKEN Scanner

T_8

	B_1	B_2	B_3	B_4	B_5	D
A_1	730,9				26	756,9
A_2	610,1					610,1
A_3	734,5					734,5
A_4	306,5		337,5	268,8		912,8
A_5	116,7					116,7
A_6		88,5		25,2		113,7
A_7	44,3					44,3
A_8					58,2	58,2
N	2 543,0	88,5	337,5	294,0	84,2	3 347,2

C. Rețele pentru planificarea, conducerea și controlul producției

§ 1. INTRODUCERE

Datorită ritmului extrem de rapid al pătrunderii științei în producție, pe de o parte, iar pe de altă parte, datorită felului cum este concepută ființa omenească, receptivă la orice schimbare, a apărut necesitatea unui simbol al producției numit „proiect".

Lipsa unor metode care să permită controlul diverselor activități pe parcursul desfășurării acestora a condus la apariția modelului de rețea al proiectului, care să permită cunoașterea desfășurării procesului de producție înainte ca acesta să fi avut loc.

Eforturile făcute în această direcție au condus la elaborarea diverselor metode și modele cunoscute sub diverse denumiri ca P.E.R.T., C.M.P. și altele.

Scanned with OKEN Scanner

La vremea sa, matematicianul Kelley s-a ocupat în mod special de metoda C.M.P., folosind programarea liniară parametrică.

Obiectivul său era însă limitat la cercetarea timpului optim al fiecărei faze a proiectului.

Mai tîrziu, asemenea încercări au atras atenția multor cercetători, datorită simplicității, eficacității și largului cîmp de aplicabilitate ale acestora.

Metodele o dată stabilite au devenit atotcuprinzătoare, aplicabile în sectoare de mare importanță, cum sînt construcțiile de obiective noi, reparațiile capitale, lansarea în producție a unui produs complex, organizarea lucrărilor de cercetare, efectuarea prospecțiunilor geologice de mari proporții, elaborarea fluxului de producție pe bandă rulantă, repartizarea sarcinilor pe mașini etc.

Din enumerarea acestor cîtorva domenii de aplicabilitate, reiese clar că obiectivul planificării și conducerii în rețea se referă la procese complexe, la care iau parte mulți executanți legați între ei prin angajamente reciproce.

§ 2. ELEMENTELE REȚELELOR DE ACTIVITATE

Mijlocul de reprezentare a metodei folosită în planificarea și conducerea operativă a unui proces este graful rețea, care modelează întregul program ce trebuie realizat.

În acest scop, considerăm proiectul format din activități reprezentate prin arce și evenimente reprezentate prin puncte.

Astfel, dacă considerăm arcul (x_i, x_j), legînd vîrful x_i de vîrful x_j, el va reprezenta activitatea ce trebuie plasată după realizarea evenimentului x_i; sensul indicat pe arce arată că activitatea x_i trebuie să preceadă activitatea x_j. Cu aceste precizări, se realizează graful rețea cu următoarele caracteristici:

a) Numerotăm vîrfurile în așa fel, încît un vîrf cu indice mai mic precede un vîrf cu indice mai mare.

b) Fiecare vîrf trebuie să aibă cel puțin un arc de intrare și unul de ieșire, cu excepția unui vîrf x_0, numit intrarea

Scanned with OKEN Scanner

în rețea (care nu are nici un arc de intrare) și a unui vîrf x_n, numit ieșirea din rețea (care nu are nici un arc de ieșire).

c) Evenimentul fiind rezultatul îndeplinirii unei activități, el trebuie să fie concret, precis și să aibă importanță esențială pentru proiectul considerat.

d) Deoarece între evenimente există o anumită ordine strictă, aceasta trebuie astfel reprezentată, încît unul nu poate avea loc înaintea celui care-l precede.

e) Fiecare lucrare începe și se termină cu un anumit eveniment; evenimentul anterior este punctul de plecare pentru o lucrare, iar evenimentul următor este rezultatul aceleiași lucrări.

Un eveniment poate fi începutul sau rezultatul mai multor lucrări.

Lucrările fiind legate prin arce, acestea arată legătura și succesiunea lor.

Succesiunea vîrfurilor este de o mare importanță, întrucît ea arată ce lucrări trebuiesc terminate înainte de a avea loc desfășurarea altora.

f) Fiecare lucrare, reprezentată printr-un arc orientat, trebuie apreciată în timp cu ajutorul unor măsurători oarecare; de obicei, pentru stabilirea duratei se folosește teoria probabilităților și statistica matematică, care ne dau durata medie a fiecărei lucrări.

Înainte de a trece la trasarea grafului rețea se alcătuiește o enumerare a lucrărilor, duratele medii și condiționarea dintre ele.

§ 3. CRITERII DE CONSTRUIRE A UNEI REȚELE DE ACTIVITATE

După ce am arătat în linii mari, care sînt elementele componente care duc la formarea conceptului de rețea, să vedem care este modul concret de realizare a acesteia.

Experiența practică a scos la iveală o serie de reguli și procedee, de care trebuie să se țină seama în construirea

Scanned with OKEN Scanner

rețelei, izvorîte din nevoi practice, pentru ca aceasta să redea în mod clar și cît mai simplu posibil imaginea concretă a obiectivului urmărit. Așa au apărut noțiunile de „componență" și de „operație fictivă", care permit să se păstreze principiul succesiunii evenimentelor și lucrărilor și care duc la o simplificare a grafului construit.

Fie, de exemplu, cunoscut faptul că lucrarea L precede lucrarea H.

În paralel cu lucrarea L, putem executa lucrările I și J, a căror execuție condiționează de asemenea începutul lucrării H; însă după executarea a $\frac{2}{5}$ din L poate începe I și după executarea a $\frac{4}{5}$ din L poate începe lucrarea J. În acest caz, lucrarea L se descompune în trei componente distincte:

L_1 — execuția a $\frac{2}{5}$ din L,

L_2 — execuția a încă $\frac{2}{5}$ din L, dar în paralel cu lucrarea I.

L_3 — execuția a $\frac{1}{5}$ din L, în paralel cu I și J.

Schematic, această descompunere se prezintă ca în figura III.C.3.1.

Dacă două lucrări — sau mai multe — trebuie să înceapă după realizarea unuia și aceluiași eveniment și să se ter-

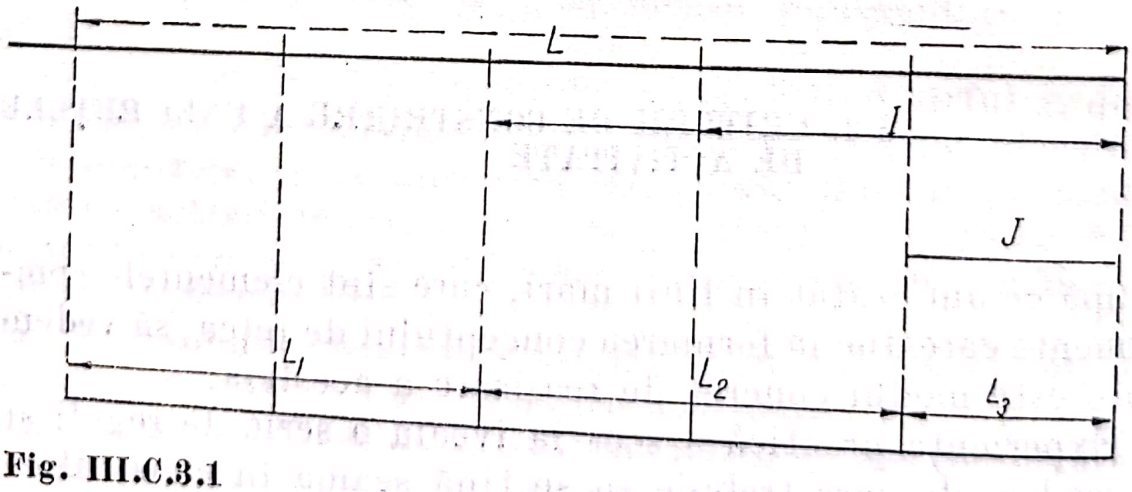


Fig. III.C.3.1

Scanned with OKEN Scanner

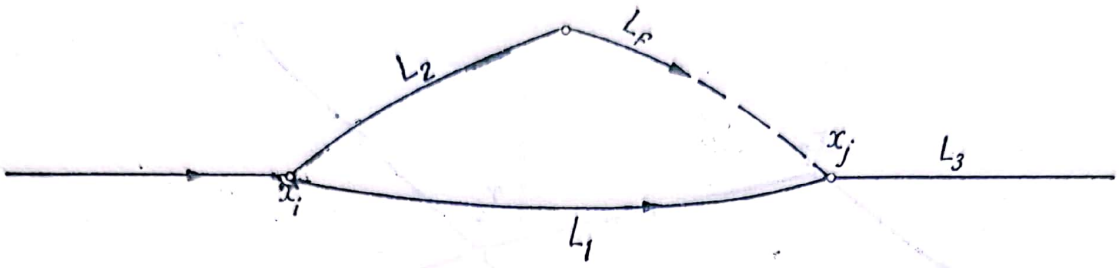


Fig. III.C.3.2

mine în același timp la evenimentul următor, notarea lor identică ar putea duce la confuzii.

S-au găsit soluții și în această situație, prin introducerea unei operații zise „fictive", a cărei durată de execuție va fi evident egală cu zero.

De exemplu, operațiile L_1 și L_2 ce trebuie executate în același timp sînt condiționate de același eveniment x_i și condiționează un același eveniment (începerea lucrării L_3), după cum rezultă din figura III.C.3.2.

Tot în scopul simplificării rețelei, recurgem adesea la studierea rețelei pe porțiuni de lucrări, care pot fi considerate ca un subproiect al proiectului dat și în care lucrările sînt strîns legate una de alta și sînt condiționate și condiționează la rîndul lor aceeași lucrare.

Evident că separarea unui grup de lucrări din totalul lucrărilor cuprinse în rețea duce la o simplificare vădită a studiului schemei generale.

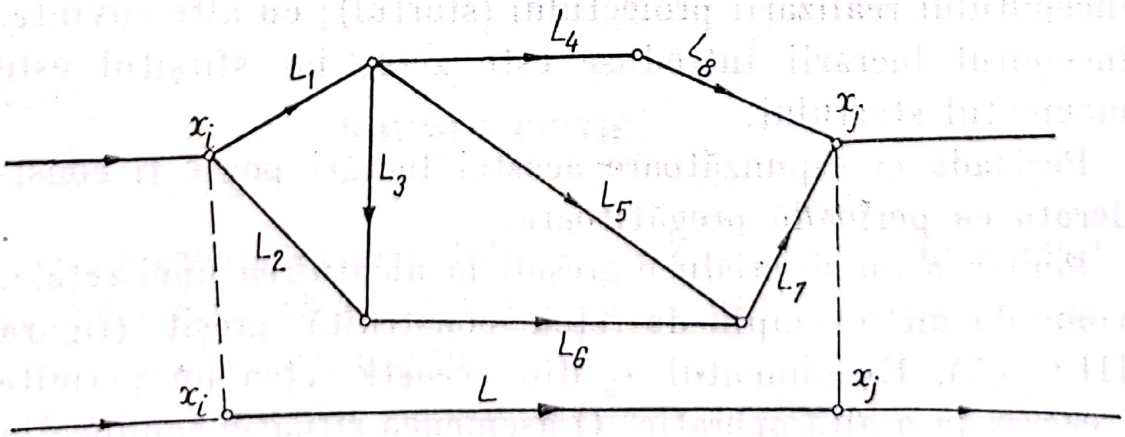


Fig. III.C.3.3

Scanned with OKEN Scanner

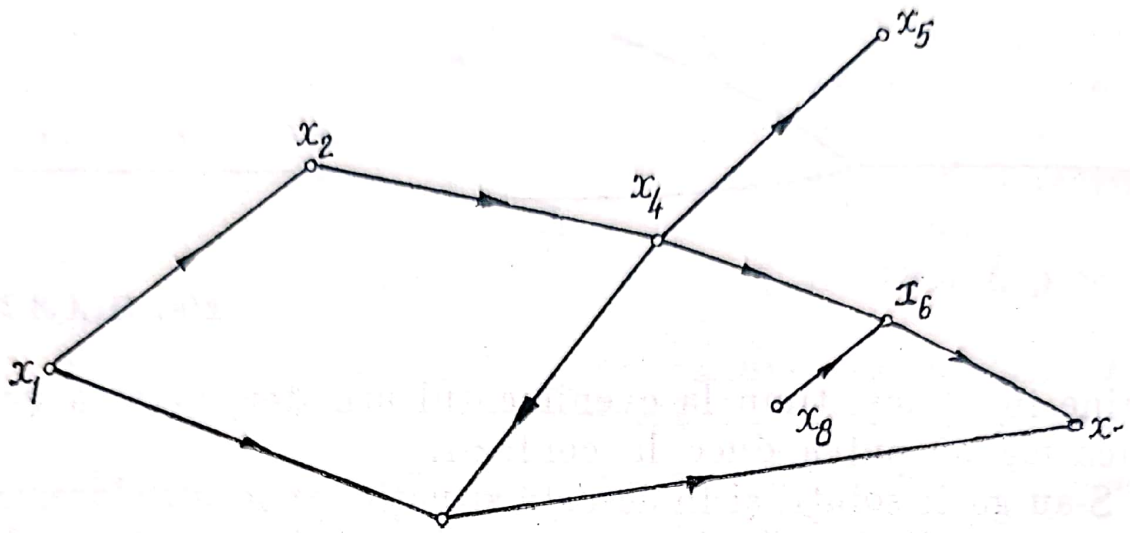


Fig. III.C.3.4

Exemplu. Grupa de lucrări $L_1, L_2, \ldots, L_8$ se analizează separat și în rețeaua inițială se înlocuiește printr-o singură lucrare L, (figura III.C.3.3).

Adesea, apar situații în care o anumită lucrare L_i urmează după două lucrări paralele L_k și L_p. După L_p, urmează L_q, care însă nu este condiționată de L_i. Pentru reprezentarea grafică a unei astfel de situații, se recurge la introducerea lucrării fictive, ca în schema din figura III.C.3.4.

De asemenea, este nevoie să se determine perioada calendaristică de elaborare a proiectului. În acest scop, se introduce o lucrare al cărei timp de execuție coincide cu data începutului realizării proiectului (startul); cu alte cuvinte, începutul lucrării introduse este zero, iar sfîrșitul este momentul startului.

Perioada corespunzătoare acestei lucrări poate fi considerată ca perioadă pregătitoare.

Pentru a nu se produce greșeli la alcătuirea unei rețele, vom da un exemplu de rețea construită greșit (figura III.C.3.5). Evenimentul x_5 din această rețea nu permite trecerea la o altă operație. O asemenea situație conduce la concluzia că s-a comis o greșeală, fie în construirea grafului,

Scanned with OKEN Scanner

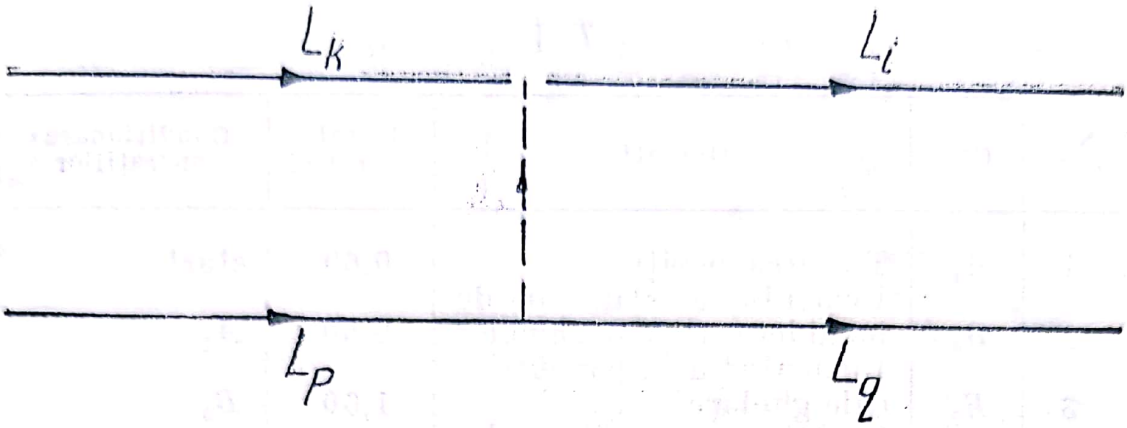


Fig. III.C.3.5

fie în faptul că lucrarea anterioară acestui eveniment nu este utilă proiectului și ca atare poate fi eliminată.

De asemenea, o analiză a vîrfului x_8 arată că acesta nu are nici un eveniment care să-l preceadă, de unde reiese construirea greșită a rețelei sau omiterea unor lucrări care condiționează începerea lucrării (x_8, x_6), ca atare trebuie determinate lucrările omise și incluse în rețea.

Este evident că în rețele nu pot apărea circuite. O astfel de situație ar conduce la greșeli de logică, în sensul că o dată executată o lucrare, ar trebui să revenim asupra ei, ceea ce este absurd.

Numai o cunoaștere temeinică și luarea în considerație a tuturor celor de mai sus pot să ducă la realizarea unei rețele corect întocmite.

§ 4. DRUMUL CRITIC

După prezentarea metodologiei de construire a unei rețele de activitate, să considerăm un proiect referitor la montarea camerelor cofrajului spațial.

Lista operațiilor corespunzătoare proiectului, condiționarea și timpii necesari de realizare sînt dați în tabelul T I.

Scanned with OKEN Scanner

T I

Nr. nod.	Cod.	Operaţii	Durata operaţiei	Condiţionarea operaţiilor
1	A_1	Trasarea poziţiei	0,60	start
2	B_3	Demontarea stîlpilor de închidere a diafragmelor	0,50	A_1
3	B_9	Demontarea panourilor cale-ghidare	1,60	B_3
4	B_3	Curăţirea panourilor cale-ghidare	1,00	B_9
5	A_2	Montare panouri cale-ghidare	2,50	B_{13}
8	C	Montare armături în diafragmă	0,60	A_{12}
9	A_3	Montare camere cofrag spaţiale	9,30	C
6	A_5	Montare panou puţ-lift	2,10	B_{13}
7	A_4	Montare panou închidere tronson	4,50	A_5
13	A_6	Montare rame panouri uşi ferestre	0,50	A_3
12	A_7	Montare stîlpi închidere diafragme	0,60	A_3
10	A_8	Montare plintelor de fund	0,20	A_3
15	A_9	Montare pieselor de colţ	0,10	A_8
14	A_{10}	Montare paziilor de închidere a plăcii	0,10	A_6 şi A_7
17	A_{11}	Montare balustrade de protecţie	0,20	A_{10}
18	A_{12}	Montare goluri tehnologice	0,15	A_{11}
19	A_{14}	Montare armături pentru placă	3,10	A_{10}
11	A_{15}	Executarea de izolaţii cu P.F.L. poros la rosturi	4,60	A_4
16	A_{13}	Astuparea lufturilor	4,00	A_5 şi A_{15}
20	D	Turnare beton	6,60	$A_9\ A_{12}\ A_{13}\ A_{14}$
21	E	Tratament termic	7,50	D
22	B_{11}	Demontarea pieselor mărunte	6,00	E

Menţionăm că duratele au fost calculate în ore zecimi pentru simplificarea calculelor, ceea ce înseamnă că 100 de minute centesimale echivalează cu 60 de minute sexagesimale.

Scanned with OKEN Scanner

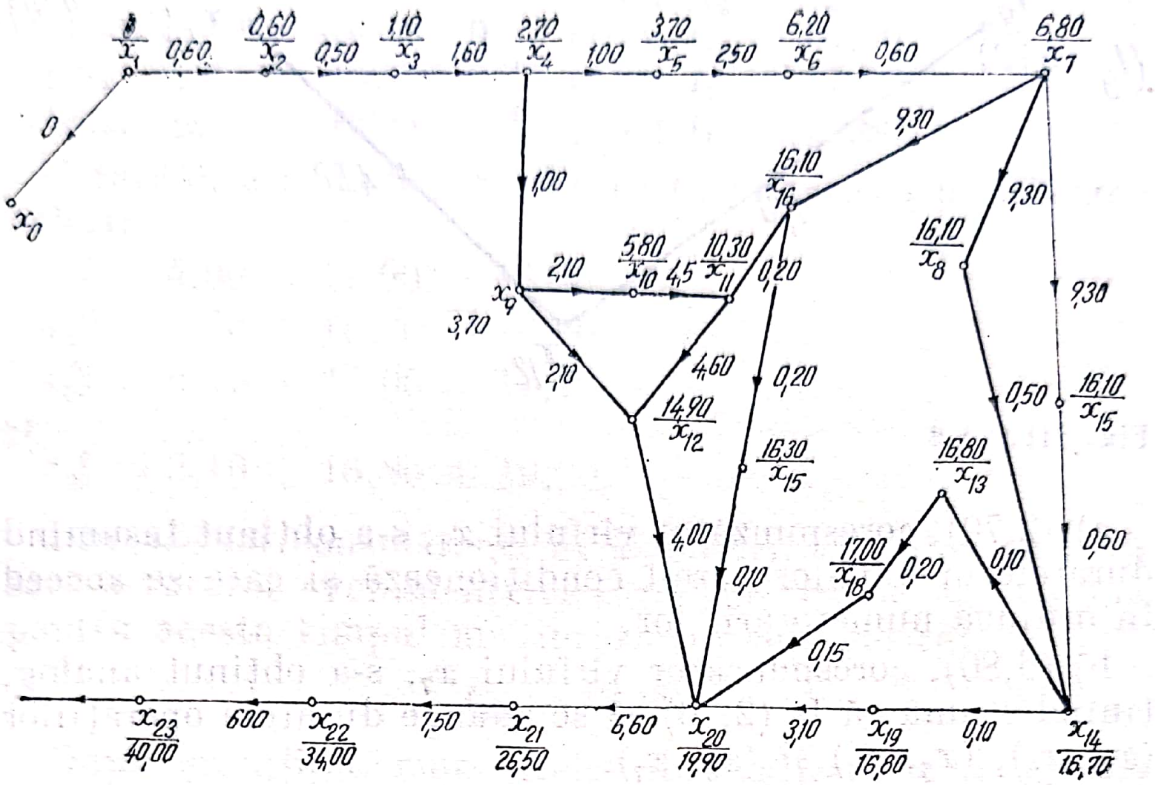


Fig. III.C.4.1

Graful corespunzător acestei lucrări, ținînd cont de succesiunea operațiilor componente, de condiționarea lor, precum și de duratele corespunzătoare, este cel din figura III.C.4.1.

Evident că o asemenea reprezentare este mult mai expresivă decît descrierea sub formă de tabel de lucrări.

În legătură cu rețeaua de mai sus punem problema determinării timpului optim de realizare a întregului proiect.

Durata drumului de parcurs, și care trebuie determinată, trebuie aleasă astfel încît să asigure realizarea întregului proiect.

Acest drum, cu durata cea mai mare posibilă, se numește „drum critic". El va reprezenta durata totală a acelor lucrări care limitează întregul proces proiectat, determină timpul calendaristic de terminare a proiectului și toate evenimentele cheie ale procesului.

Astfel, în graful din figura III.C.4.1. drumul critic trasat cu linie subțire s-a stabilit după cum urmează:

Scanned with OKEN Scanner

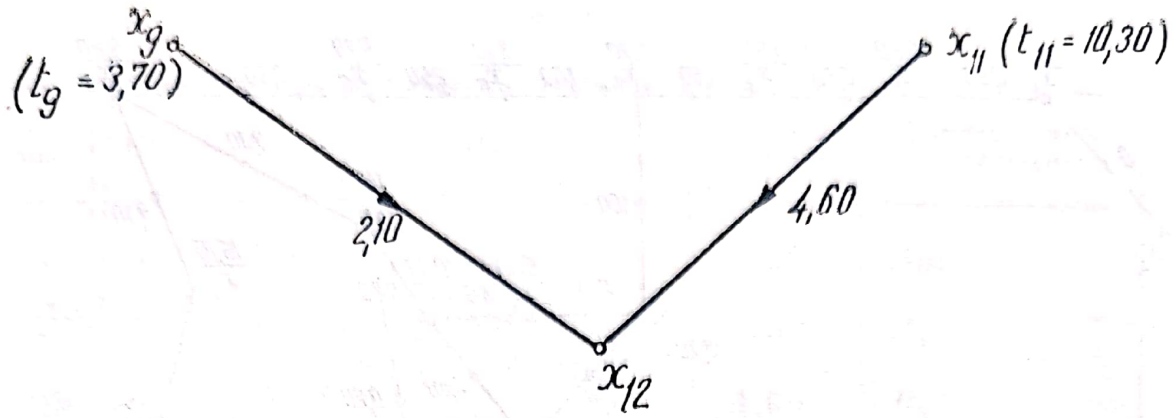


Fig. III.C.4.2

a) (2,70), corespunzător vîrfului x_4, s-a obținut însumînd duratele operațiilor care-l condiționează și care se succed în ordinea numerotării lor.

b) (6,80), corespunzător vîrfului x_7, s-a obținut analog, ținînd seama că la (2,70) să se adauge duratele operațiilor (x_4, x_5), (x_5, x_6) și (x_6, x_7).

c) Vîrful (x_7) precede vîrfurile x_8, x_{16}, și x_{15}; pentru obținerea timpilor de realizare a acestor evenimente adăugăm la (6,80) durata operației corespunzătoare, obținîndu-se valorile deja înregistrate pe graf ș.a.m.d.

În situația evenimentului x_{12}, care este condiționat de operațiile (x_{11}, x_{12}), cu durata (4,60) și (x_9, x_{12}), cu durata 2,10, sîntem obligați să alegem una dintre cele două posibilități ilustrate în figura III.C.4.2.

Ținînd seama de prima condiționare, evenimentului x_{12} îi corespunde momentul de realizare,

$$t'_{12} = 10,30 + 4,60 = 14,90,$$

în timp ce datorită celei de-a doua condiționări, îi corespunde

$$t''_{12} = 3,70 + 2,10 = 5,80$$

Evident că dintre aceste două rezultate vom lua în considerație pe acela care corespunde momentului maxim $t_{12} = 14,90$, deoarece numai în felul acesta se poate asigura realizarea operațiilor care preced pe x_{12}, prin x_{11}.

De asemenea, după cum se vede, evenimentul x_{20} este condiționat de operațiile:

(x_{12}, x_{20}), cu durata 4,00 și cu $t_{12} = 14{,}90$,
(x_{17}, x_{20}), cu durata 0,10 și cu $t_{17} = 16{,}30$,
(x_{18}, x_{20}), cu durata 0,15 și cu $t_{18} = 17{,}00$,
(x_{19}, x_{20}), cu durata 3,10 și cu $t_{19} = 16{,}80$.

Corespunzător acestora, pentru t_{20} ar rezulta următoarele valori:

$$t_{20}^{(1)} = 4{,}00 + 14{,}90 = 18{,}90,$$
$$t_{20}^{(2)} = 0{,}10 + 16{,}30 = 16{,}40,$$
$$t_{20}^{(3)} = 0{,}15 + 17{,}00 = 17{,}15$$

și

$$t_{20}^{(4} = 3{,}10 + 16{,}80 + 19{,}90.$$

Din același considerent, al asigurării timpilor de realizare a tuturor evenimentelor care-l preced pe x_{20}, s-a luat pentru acesta timpul maxim dintre cele găsite:

$$t_{20} = 19{,}90.$$

După stabilirea momentelor de realizare t_i a tuturor evenimentelor $x_i \in X$, s-au luat în considerație acelea care au determinat timpul maxim de 40 de unități pentru x_{23}, și anume: x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_{15}, x_{14}, x_{19}, x_{20}, x_{21}, x_{22}, x_{23}, care constituie drumul critic în rețea.

Drumul critic, din graful considerat, s-a putut determina cu ușurință din aproape în aproape, deoarece proiectul conține un număr mic de operații. Evident că în cazul unui proiect mai complex se impune folosirea unor metode mai eficace, care să permită folosirea mașinilor electronice.

Același drum critic, fiind drumul maxim în graf, se poate găsi folosind și algoritmul Bellman-Kalaba, calculele fiind date în tabelul T.

Noțiunea de „drum critic“ este de o deosebită importanță, în sistemul planificării în rețea.

Toate lucrările situate pe acest drum sînt strict încadrate în timp, ceea ce nu permite nici o întîrziere în executarea lor (fie ea impusă chiar de condiții obiective) impunînd executanților să-și concentreze toată atenția asupra acestora.

Trebuie făcută o deosebire netă între lucrările principale (așezate pe drumul critic) și lucrările secundare (nesituate pe drumul critic), care dispun de anumite rezerve de timp. Astfel, la nevoie este posibil să se facă transferul de forță

Scanned with OKEN Scanner

și mijloace de muncă afectate lucrărilor secundare la lucrările principale, în scopul reducerii timpului necesar realizării întregului proiect. Acest transfer se impune în două situații distincte, și anume:

— fie în cazul în care termenul de realizat a lucrării, rezultat din calcule, depășește termenul stabilit prin directive;

— fie în cazul în care, datorită defecțiunilor tehnice sau organizatorice, au loc întîrzieri ale lucrărilor principale.

Operațiile nesituate pe drumul critic oferă posibilitatea unor rezerve de timp și o elasticitate în executarea lor.

Cunoașterea rezervei de timp pentru toate lucrările procesului proiectat reprezintă încă o latură pozitivă a sistemului de conducere și planificare în rețea.

Cele de mai sus arată că găsirea drumului critic este un mijloc eficace nu numai pentru planificarea procesului, ci și pentru dirijarea lui.

§ 5. REZERVE DE TIMP ÎN ELABORAREA PROIECTELOR

Ținînd seamă de importanța rezervelor de timp, este necesar să lămurim unele noțiuni importante și să dăm unele definiții.

Să considerăm o activitate oarecare într-o rețea (figura III.C.5.1) unde am notat prin:

$T_t^{(i)}$, timpul cel mai lung de realizare a evenimentului x_1,

$T_d^{(i)}$, timpul cel mai scurt de realizare a evenimentului x_1.

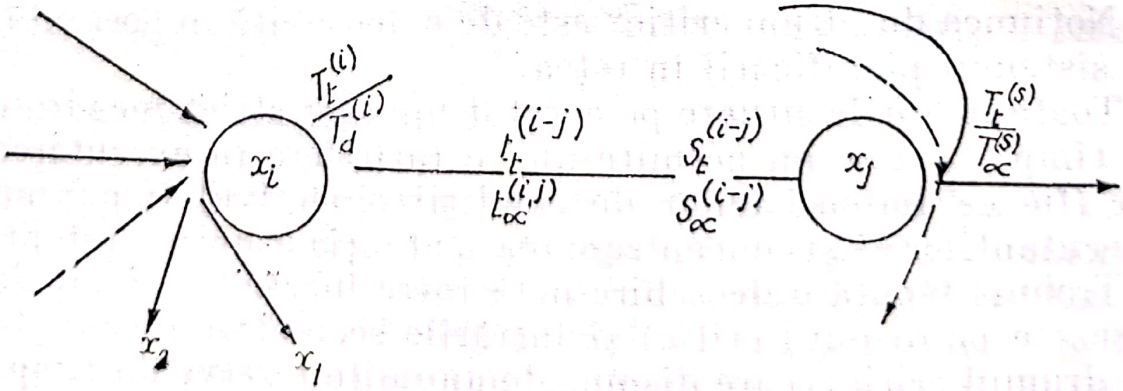


Fig. III.C.5.1

Scanned with OKEN Scanner

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}	x_{21}	x_{22}	x_{23}
x_0	0	0	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞
x_1	−∞	0	0,60	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞
x_2	−∞	−∞	0	0,50	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞
x_3	−∞	−∞	−∞	0	1,60	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	− ∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞
x_4	−∞	−∞	−∞	−∞	0	1,00	−∞	−∞	−∞	1,00	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞
x_5	−∞	−∞	−∞	−∞	−∞	0	2,50	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞
x_6	−∞	−∞	−∞	−∞	−∞	−∞	0	0,60	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞
x_7	−∞	−∞	−∞	−∞	−∞	−∞	−∞	0	9,30	−∞	−∞	−∞	−∞	−∞	−∞	9,30	−∞	9,30	−∞	−∞	−∞	−∞	−∞	−∞
x_8	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	0	−∞	−∞	−∞	−∞	−∞	0,50	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞
x_9	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	0	2,10	−∞	2,10	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞
x_{10}	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	0	4,50	—	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞
x_{11}	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	0	4,60	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞
x_{12}	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	0	−∞	−∞	−∞	−∞	−∞	−∞	4,00	−∞	−∞	−∞	−∞
x_{13}	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	0	−∞	−∞	−∞	−∞	0,20	−∞	−∞	−∞	−∞	−∞
x_{14}	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	0,10	0	−∞	0,10	−∞	−∞	−∞	−∞	−∞	−∞	−∞
x_{15}	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	0,60	0	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞
x_{16}	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	0	0,20	−∞	−∞	3,10	−∞	−∞	−∞	−∞
x_{17}	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	0	−∞	−∞	0,10	−∞	−∞	−∞	−∞
x_{18}	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	0	−∞	0,15	−∞	−∞	−∞	−∞
x_{19}	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	0	3,10	−∞	−∞	−∞	−∞
x_{20}	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	0	6,60	−∞	−∞	−∞
x_{21}	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	0	7,50	−∞	−∞
x_{22}	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	0	6,00	−∞
x_{23}	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	0	−∞

Scanned with OKEN Scanner

x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}	x_{21}	x_{22}	x_{23}	$V_i^{(0)}$	$V_i^{(1)}$	$V_i^{(2)}$	$V_i^{(3)}$	$V_i^{(4)}$	$V_i^{(5)}$	$V_i^{(6)}$	$V_i^{(7)}$	$V_i^{(8)}$	$V_i^{(9)}$	$V_i^{(10)}$	$V_i^{(11)}$	$V_i^{(12)}$	$V_i^{(13)}$
−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	—	—	—	—	—	40	40
−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	—	—	—	—	—	40	40
−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	—	—	—	—	3940	3940	3940
∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	—	—	—	3890	3890	3890	3890
∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	—	—	3730	3730	3730	3730	3730
∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	—	3630	3630	3630	3630	3630	3630
0	−∞	9,30	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	33,80	3380	3380	3380	3380	3380	3380	3380
∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	29,50	33,20	33,20	3320	3320	3320	3320	3320	3320	3320
∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	23,80	23,80	2380	2380	2380	2380	2380	2380	2380
	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	26,20	26,20	2620	2620	2620	2620	2620	2620	2620
	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	31,20	3120	3120	3120	3120	3120	3120	3120
	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	26,70	26,70	2670	2670	2670	2670	2670	2670	2670
	−∞	−∞	−∞	−∞	4,00	−∞	−∞	−∞	−∞	−∞	−∞	24,10	24,10	24,10	24,10	2410	2410	2410	2410	2410	2410	2410
	−∞	−∞	0,20	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	20,45	20,45	20,45	2045	2045	2045	2045	2045	2045	2045
0,10	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	23,30	23,30	23,30	2330	2330	2330	2330	2330	2330	2330
−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞	23,90	23,90	2390	2390	2390	2390	2390	2390	2390
)	0,20	−∞	−∞	3,10	−∞	−∞	−∞	−∞	−∞	−∞	−∞	23,20	23,20	23,20	23,20	2320	2320	2320	2320	2320	2320	2320
−∞	0	−∞	−∞	0,10	−∞	−∞	−∞	−∞	−∞	−∞	−∞	20,20	20,20	20,20	20,20	2020	2020	2020	2020	2020	2020	2020
−∞	−∞	0	−∞	0,15	−∞	−∞	−∞	−∞	−∞	−∞	−∞	20,25	20,25	20,25	20,25	2025	2025	2025	2025	2025	2025	2025
−∞	−∞	−∞	0	3,10	−∞	−∞	−∞	−∞	−∞	−∞	−∞	23,20	23,20	23,20	23,20	2320	2320	2320	2320	2320	2320	2320
−∞	−∞	−∞	−∞	0	6,60	−∞	−∞	−∞	−∞	−∞	20,10	20,10	20,10	20,10	20,10	2010	2010	2010	2010	2010	2010	2010
−∞	−∞	−∞	−∞	−∞	0	7,50	−∞	−∞	−∞	13,50	13,50	13,50	13,50	13,50	13,50	1350	1350	1350	1350	1350	1350	1350
−∞	−∞	−∞	−∞	−∞	−∞	0	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6,00	6	6	6	6	6	6	6
−∞	−∞	−∞	−∞	−∞	−∞	−∞	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Scanned with OKEN Scanner

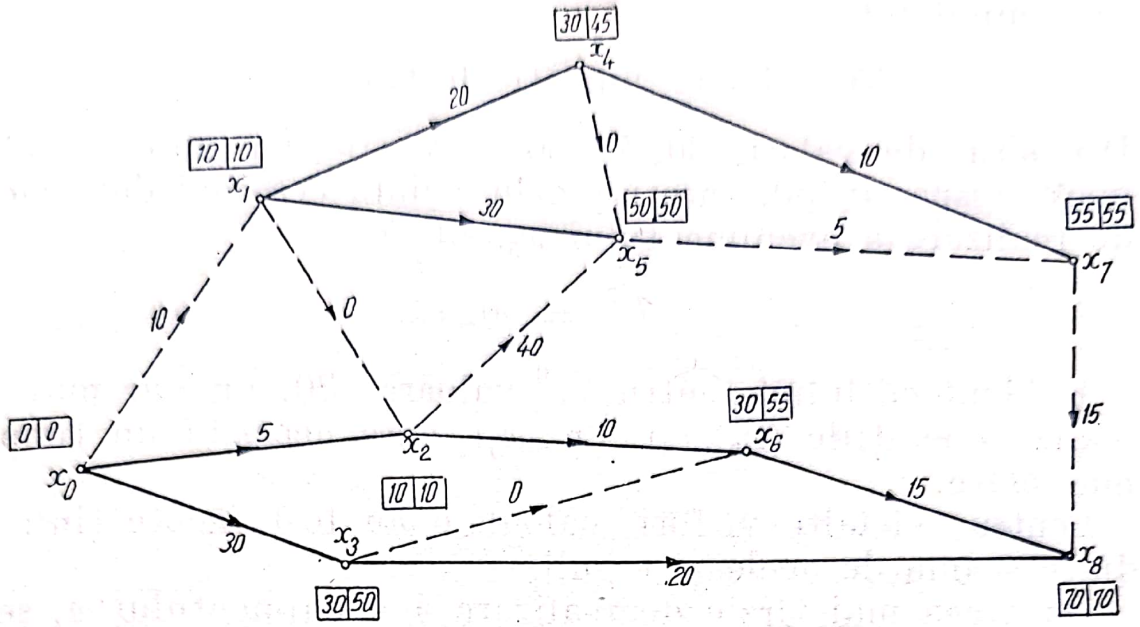


Fig. III.C.5.2

Timpul cel mai scurt de realizare a evenimentului x_1, $T_d^{(i)}$, este reprezentat de drumul maxim de la vîrful x_0 (intrarea în rețea) la vîrful x_1 considerat.

Ca exemplu, considerăm figura III.C.5.2.

Pentru înțelegerea modului cum au fost calculați timpii cei mai scurți de realizare a evenimentului x_i, ne vom referi la evenimentul x_5.

Evenimentul x_5 se realizează dacă s-au efectuat activitățile:

$$(x_0, x_1) + (x_1, x_4) + (x_4, x_5)$$

cu timpul total

$$10 + 20 + 0 = 30 \text{ de unități de timp;}$$

$$(x_0, x_1) + (x_1, x_5)$$

cu timpul total

$$10 + 30 = 40 \text{ de unități de timp;}$$

$$(x_0, x_1) + (x_1, x_8) + (x_8, x_5)$$

cu timpul total

$$10 + 0 + 40 = 50 \text{ de unități de timp;}$$

și

$$(x_0, x_8) + (x_8, x_5)$$

Scanned with OKEN Scanner

cu timpul total

$$5 + 40 = 45 \text{ de unități de timp.}$$

Din șirul de valori, 30, 40, 50 și 45 alegem pe cea mai mare, respectiv 50, care reprezintă data cea mai devreme de realizare a evenimentului x_5, adică

$$T_d^{(5)} = 50.$$

Evident că luînd pentru $T_d^{(5)}$ valoarea 30, nu s-ar putea realiza operațiile (x_0, x_8), (x_8, x_5), care necesită un timp mai mare.

Pentru celelalte vîrfuri, calculele au fost făcute ținîndu-se seama de aceleași reguli.

Data cea mai tîrzie de realizare a evenimentului x_i se calculează din aproape în aproape, pornind de la data finală de realizare a întregului proiect, cu condiția ca să fie asigurate toate operațiile care pleacă din x_i și să nu fie depășit termenul minim de realizare a întregii lucrări.

Astfel, pe graful din figura III.C.5.2 numerele $T_t^{(i)}$, trecute în partea dreaptă a numerelor $T_d^{(i)}$, din dreptunghiurile atașate vîrfurilor, se calculează după cum urmează:

$$T_t^{(8)} = T_d^{(8)} = 70,$$
$$T_t^{(7)} = T_d^{(8)} - t_{78} = 70 - 15 = 55,$$
$$T_t^{(6)} = T_d^{(8)} - t_{68} = 70 - 15 = 55,$$
$$T_t^{(3)} = T_d^{(8)} - t_{38} = 70 - 20 = 50,$$
$$T_t^{(5)} = T_t^{(7)} - t_{57} = 55 - 5 = 50,$$
$$T_t^{(4)} = \min T_t^{(5)} - t_{45};\ T_t^{(7)} - t_{47} = \min \{ 50 - 0 ;$$
$$55 - 10\} = \min \{ 50;\ 45 \} = 45.$$

Dacă am lua drept dată cea mai tîrzie de realizare a evenimentului x_4.

$$T_t^{(4)} = 50,$$

aceasta ar duce la o întîrziere cu 5 unități de timp a realizării întregului proiect.

Într-adevăr, evenimentul x_4 realizîndu-se la momentul 50 și ținînd seama că operațiile (x_4, x_7) și (x_7, x_8) au duratele

Scanned with OKEN Scanner

respectiv de 10 și 15 unități de timp, evenimentul x_8 s-ar realiza la momentul

$$T_d^{(8)} = T_t^{(8)} = 50 + 10 + 15 = 75.$$

$$T_t^{(2)} = \min \{ T_t^{(5)} - t_{25}; T_t^{(6)} - t_{26} \} = \min \{ 50 - 40; 55 - 10 \} = \min \{ 10, 45 \} = 10,$$

$$T_t^{(1)} = \min \{ T_t^{(4)} - t_{14}; T_t^{(5)} - t_{15}; T_t^{(2)} - t_{12} \} = \min \{ 45 - 20; 50 - 30; 10 - 0 \} = \min \{ 25, 20; 10 \} = 10, \quad T_t^{(0)} = T_d^{(0)} = 0.$$

Dacă notăm cu $t_d^{(i-j)}$ timpul cel mai scurt de începere a activității (x_i, x_j) și cu $t_t^{(i-j)}$ timpul cel mai lung de începere a aceleiași activități și dacă ținem seama de faptul că activitatea (x_i, x_j) nu poate începe decît după realizarea evenimentului x_i, avem următoarele egalități utile în calcule:

$$\begin{cases} T_d^{(i)} = t_d^{(i-j)}, \\ T_t^{(i)} = t_t^{(i-j)}. \end{cases}$$

În continuare, vom nota cu $S_d^{(i-j)}$ timpul cel mai scurt de terminare a activității (x_i, x_j) și cu $S_t^{(i-j)}$ timpul cel mai lung de terminare a aceleiași activități.

Nu este greu de înțeles că între momentul cel mai îndepărtat de realizare a evenimentului x_j și momentul cel mai îndepărtat de terminare a activității (x_i, x_j) există egalitatea:

$$T_t^{(j)} = S_t^{(i-j)}.$$

Ținînd seama de semnificația notațiilor introduse, au loc următoarele relații evidente:

a) $S_d^{(i-j)} = t_d^{(i-j)} + t_{ij}$,

b) $S_t^{(i-j)} = t_t^{(i-j)} + t_{ij}$,

c) $S_d^{(i-j)} = T_d^{(i)} + t_{ij}$,

d) $t_t^{(i-j)} = T_t^{(j)} - t_{ij}$,

unde am notat cu t_{ij} durata operației (x_i, x_j).

Scanned with OKEN Scanner

Relațiile a, b, c, d pot fi urmărite cu ușurință și pe figura III.C.5.3.

Să exemplificăm cele de mai jos pe o porțiune dintr-o rețea, considerînd vîrfurile x_1, x_3, x_5, x_7 și x_9 (figura III.C.5.4).

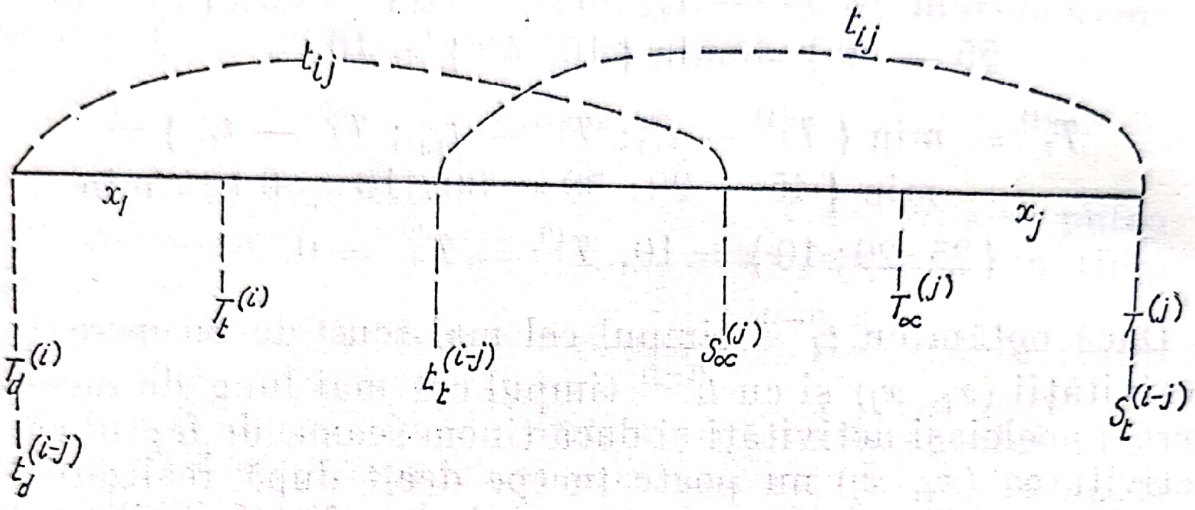


Fig. III.C.5.3

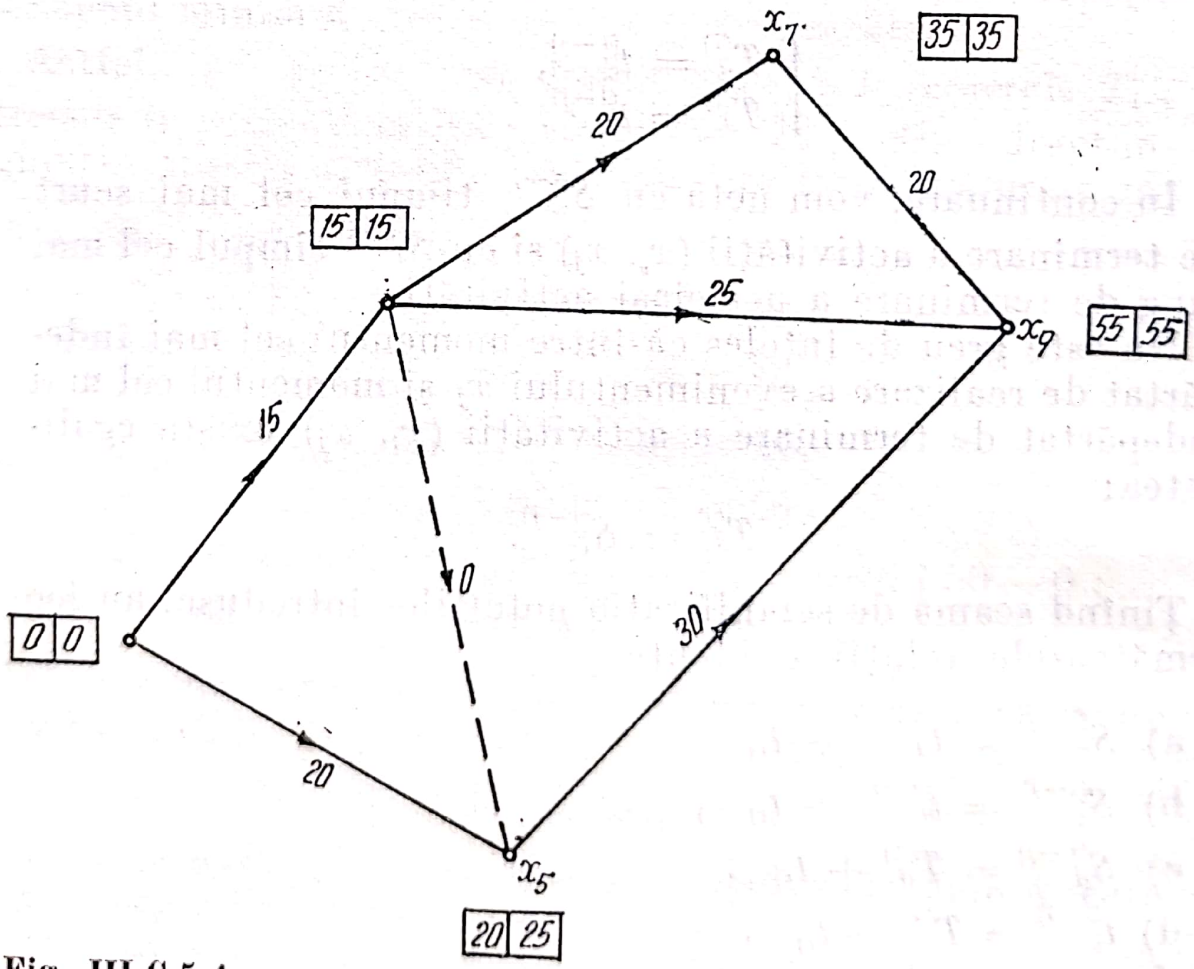


Fig. III.C.5.4

Calculele care au condus la valorile trecute în dreptunghiurile atașate vîrfurilor sînt redate în tabelul de mai jos:

x_i	x_j	Durata operației (x_i, x_j)	Momentul cel mai devreme al operației (x_i, x_j)	Sfîrșitul cel mai devreme al operației (x_i, x_j)	Timpul cel mai tîrziu de începere a operației (x_i, x_j)	
		t_{ij}	$t_d^{(i-j)} = T_d^i$	$S_d^{(i-j)} = T_d^j + t_{ij} = t_d^{(i-j)} + t_{ij}$	$t_t^{(i-j)} = T_t^{(j)} - t_{ij} = S_t^{(i-j)} - t_{ij}$	$S_t^{(i-j)}$
x_1	x_3	15	0	15	0	15
x_1	x_5	20	0	20	5	25
x_3	x_5	0	15	15	25	25
x_3	x_7	20	15	35	15	35
x_3	x_9	25	15	40	30	55
x_5	x_9	30	20	50	25	55
x_7	x_9	20	35	55	35	55

Analizînd cu atenție figura III.C.5.2., constatăm că pentru evenimentele situate pe drumul critic timpul cel mai scurt și timpul cel mai lung sînt aceleași.

Pentru celelalte operații, între acești timpi există și diferențe diferite de zero.

Această situație sugerează ideea unei posibilități de manevrare mai elastică a activităților secundare. Tocmai acest fapt a condus la noțiunea de „marjă" în activitatea de planificare și conducere.

În continuare, vom prezenta două tipuri distincte de marje: marjă liberă și marjă totală.

Scanned with OKEN Scanner

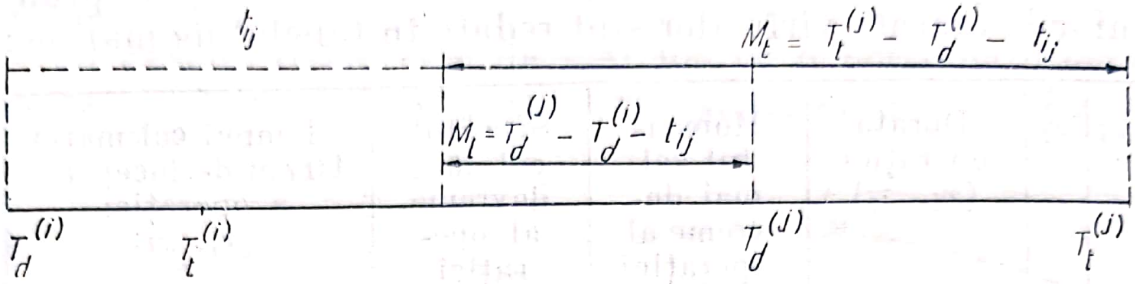


Fig. III.C.5.5

Numim „marjă liberă" M_l diferența dintre timpul cel mai scurt de realizare a evenimentului x_j și timpul cel mai lung de realizare a evenimentului x_i, mai puțin durata operației (x_i, x_j), adică

$$M_l = T_d^{(j)} - T_d^{(i)} - t_{ij}.$$

Marja totală M_t este dată de diferența dintre timpul cel mai tîrziu de realizare a evenimentului x_j și timpul cel mai devreme de realizare a evenimentului x_i, minus durata operației (x_i, x_j), adică

$$M_t = T_t^{(j)} - T_d^{(i)} - t_{ij}.$$

Să ilustrăm cele două definiții pe baza schiței din figura III.C.5.5.

Pe schiță putem vedea că marja liberă reprezintă acel interval în care poate începe operația (x_i, x_j), astfel încît operația următoare (x_i, x_k) să aibă posibilitatea să înceapă la momentul cel mai devreme. Marja totală ne dă intervalul în care se poate situa începutul operației (x_i, x_j), fără a condiționa începerea operației următoare în momentul ei cel mai devreme.

Evident că pentru operațiile de pe drumul critic, marjele sînt egale cu zero.

Reluînd graful din figura III.C.5.2., marjele totale și libere sînt date în tabelul *T* II.

Scanned with OKEN Scanner

x_i	x_j	t_{ij}	$T_d^{(i)}$	$T_d^{(j)}$	$T_l^{(j)}$	$M_l = T_d^{(j)} - T_d^{(i)} - t_{ij}$	$M_l = T_d^{(j)} - T_d^{(i)} - t_{ij}$
x_0	x_1	10	0	10	10	0	0
x_0	x_2	5	0	10	10	5	5
x_0	x_3	30	0	30	50	0	20
x_1	x_2	0	10	10	10	0	0
x_1	x_4	20	10	30	45	0	15
x_1	x_5	30	10	50	50	10	10
x_2	x_5	40	10	50	50	0	0
x_2	x_6	10	10	30	55	10	35
x_3	x_6	0	30	30	55	0	25
x_3	x_8	20	30	70	70	20	20
x_4	x_7	10	30	55	55	15	15
x_4	x_5	0	30	50	50	20	20
x_5	x_7	5	50	55	55	0	0
x_6	x_8	15	30	70	70	25	25
x_7	x_8	15	55	70	70	0	0

§ 6. GĂSIREA DRUMULUI CRITIC ŞI A REZERVELOR DE TIMP LEGATE, DE PROIECTUL DE CONSTRUIRE A UNUI IMOBIL

Lista activităţilor (figurile III.C.6.1 şi III.C.6.2):

1. Trasarea terenului.
2. Excavarea fundaţiei.
3. Excavarea conductei de apă pînă la clădire.
4. Montarea cofrajelor.
5. Turnarea fundaţiei;
6. Zidirea pereţilor.
7. Instalaţia conductei de apă pînă la intrarea în casă.
8. Montarea acoperişului.
9. Instalarea conductelor de apă la subsol.
10. Prepararea dulgheriei.
11. Instalaţia electrică.
12. Finisarea interiorului (tencuială, vopsitorie, zugrăveală).

Scanned with OKEN Scanner

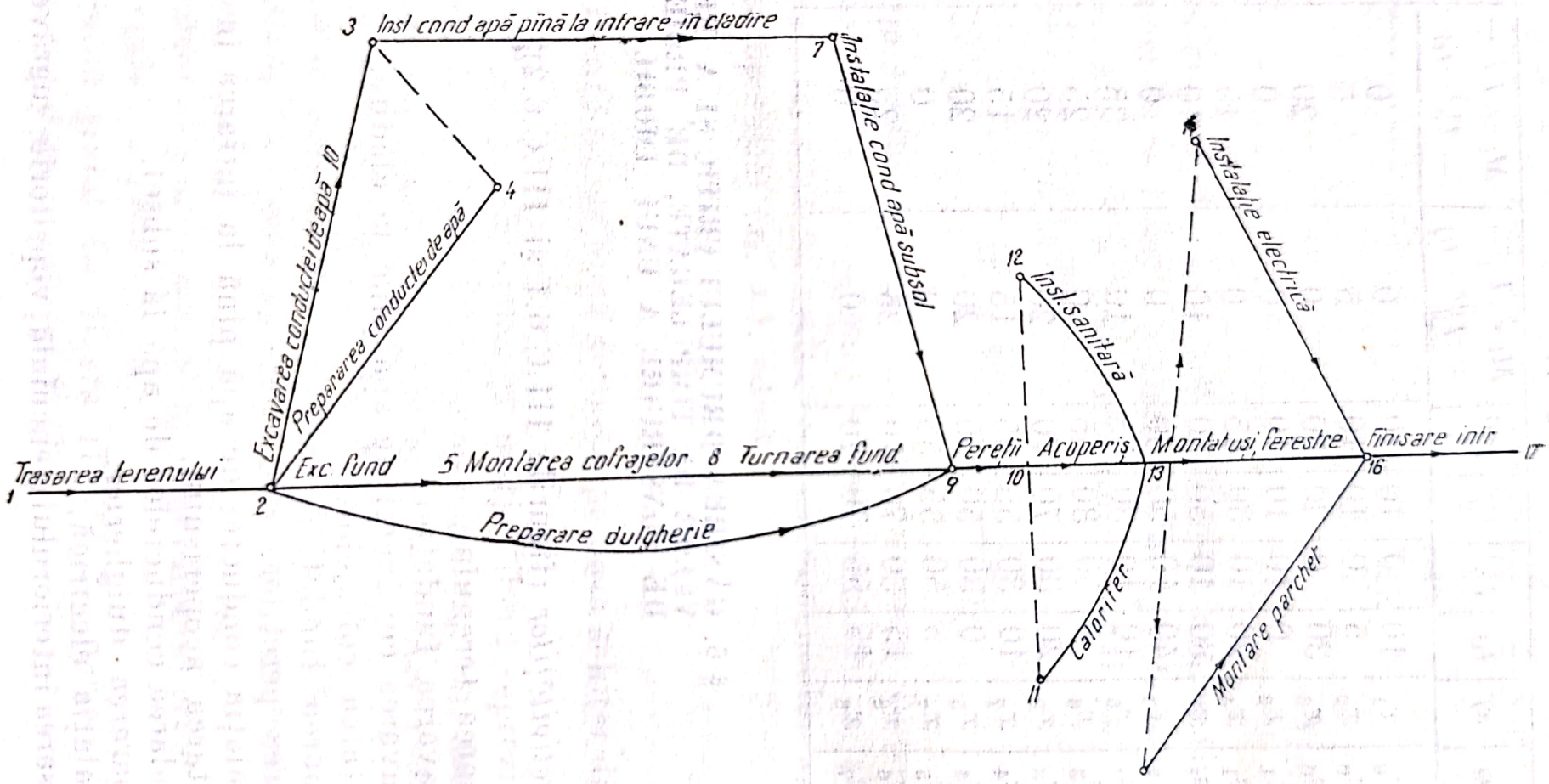
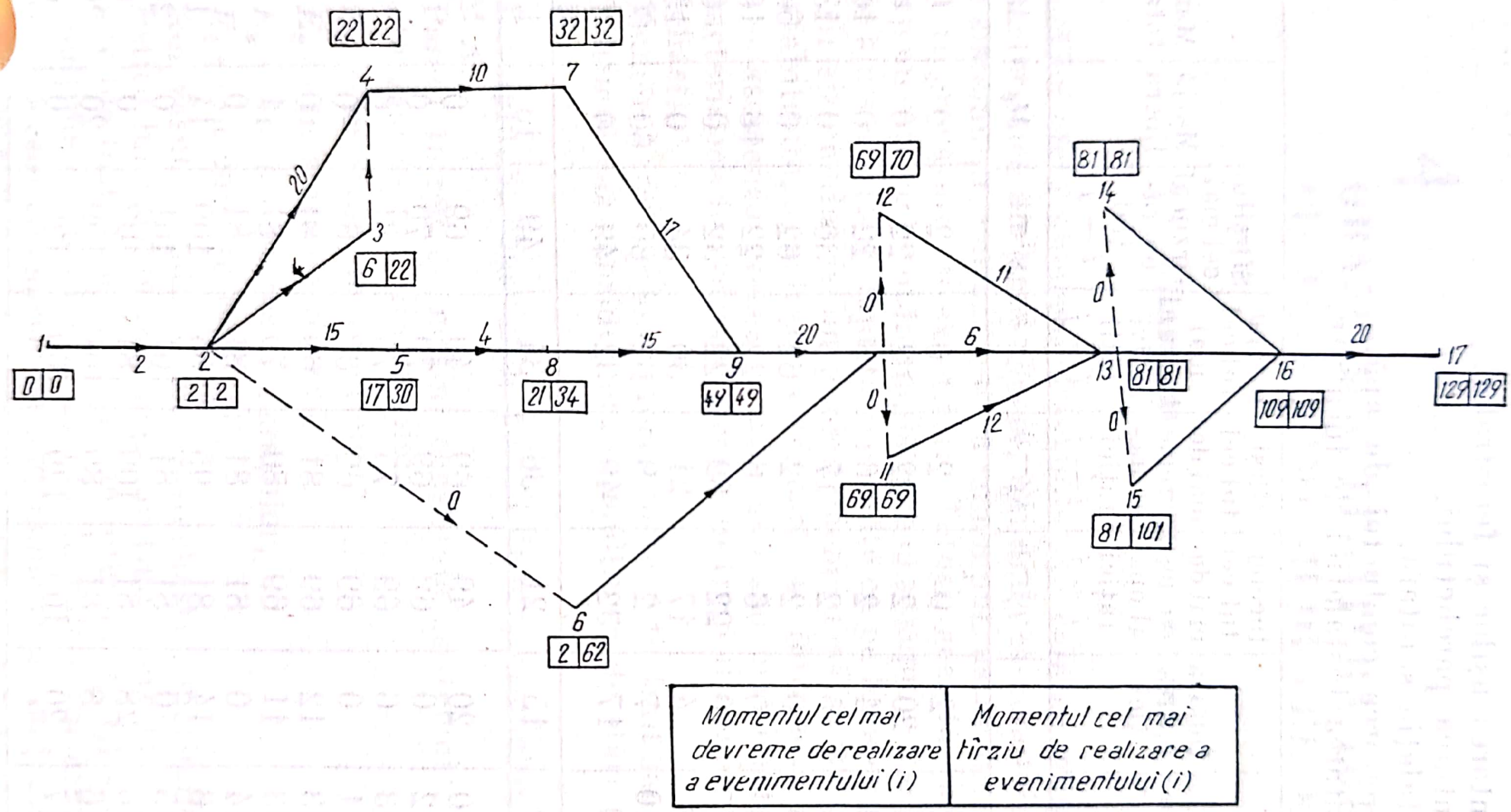

3 Insl cond apă pînă la intrare în clădire
7 Instalațiile cond apă subsol
Excavarea conductei de apă 10
Prepararea conductei de apă
4
12 Insl. sanitară
Instalațiile electrică
Trasarea terenului
1
Exc fund
2
5 Montarea cofrajelor
8 Turnarea fund.
9
Pereții
10
Acoperiș
Montat uși ferestre
13
Finisare intr
16
17
Preparare dulgherie
Caloriferr
11
Montare parchet

Fig. III.C.6.1



22 22
32 32
4
10
7
20
3
6 22
17
69 70
12
81 81
14
11
0
1
2
2
15
5
4
8
15
9
20
6
13
81 81
16
20
17
0 0
2 2
17 30
21 34
49 49
0
12
0
109 109
129 129
11
69 69
15
81 101
0
6
2 62
Momentul cel mai devreme de realizare a evenimentului (i)
Momentul cel mai tîrziu de realizare a evenimentului (i)

Fig. III.C.6.2

13. Montarea ușilor și ferestrelor.
14. Montarea parchetului.
15. Instalația sanitară.
16. Prepararea conductei de apă.
17. Montarea instalației de încălzire.

Operația	Durata operației	Începutul cel mai devreme al operației $i-j$	Sfîrșitul cel mai devreme al operației $i-j$	Începutul cel mai tîrziu al activității $i-j$	Sfîrșitul cel mai tîrziu al operației $i-j$	Marjă liberă	Marjă totală
$i-j$	t	$t_d^{(i-j)}$	$S_d^{(i-j)}$	$t_t^{(i-j)}$	$S_t^{(i-j)}$	M_l	M_t
1—2	2	0	2	0	2	0	0
2—4	20	2	22	2	22	0	0
2—3	4	2	6	18	22	0	16
2—5	15	2	17	15	30	0	13
2—6	0	2	2	62	62	0	60
3—4	0	6	6	22	22	16	16
4—7	10	22	32	22	32	0	0
5—8	4	17	21	30	34	0	13
6—10	7	2	9	62	69	60	60
7—9	17	32	49	32	49	0	0
8—9	15	21	36	34	49	13	13
9—10	20	49	69	49	69	0	0
10—12	0	69	69	70	70	0	1
10—13	6	69	75	75	81	6	6
10—11	0	69	69	69	69	0	0
11—13	12	69	81	69	81	0	0
12—13	11	69	80	70	81	1	1
13—14	0	81	81	81	81	0	0
13—16	14	81	95	95	109	14	14
13—15	0	81	81	101	101	0	20
14—16	28	81	109	81	109	0	0
15—16	8	81	89	101	109	20	20
16—17	20	109	129	109	129	0	0

Scanned with OKEN Scanner

CAPITOLUL

4

GRAFELE ŞI LEGĂTURA LOR CU ALTE DOMENII DE CERCETARE

§ 1. GRAFE ŞI LANŢURI MARKOV

Practica oferă numeroase exemple, în care anumite valori caracteristice ale unui sistem, formînd aşa-numitele „stări" ale sistemului, variază o dată cu timpul, astfel încît ele nu pot fi prevăzute cu exactitate. Totuşi, această variaţie în timp nu este cu totul întîmplătoare ci se supune legilor generale ale statisticii, fapt care permite ca pe baza cunoaşterii stării sistemului la un anumit moment să se prevadă, în probabilitate, evoluţia ulterioară a întregului sistem.

Un asemenea proces în care una sau mai multe valori caracteristice lui variază aleator în timp îl numim „proces stochastic".

Numărul stărilor unui proces poate să fie finit sau infinit, în ultimul caz considerîndu-se mulţimea de stări numărabilă.

Exemplu. Să considerăm parcul de maşini-unelte al unei I.M.A.

Parcul întreg de maşini constituie „sistemul", care se poate găsi în una din stările:

E_1 — în stare de funcţionare,

E_2 — în stare de rezervă,

E_3 — în aşteptarea reparaţiilor,

E_4 — în curs de reparaţii.

Am văzut că într-un proces stochastic, mărimile caracteristice lui variază în funcţie de timp, care la rîndul său poate lua toate valorile de pe un interval (t_1, t_2) sau numai o mulţime cel mult numărabilă de valori; în acest ultim caz, procesul stochastic se mai numeşte şi lanţ aleator.

Scanned with OKEN Scanner

Întrucît scopul nostru nu este de a face o prezentare a lanțurilor aleatoare în toată generalitatea lor, ne vom referi pe scurt la un caz particular al acestora, și anume la lanțurile de tip Markov.

Lanțul aleator de tip Markov este un șir de variabile aleatoare care satisface condiția lui Markov și anume: probabilitatea ca sistemul la momentul $(n+1)$ să se găsească în starea (i_n+1), condiționată de faptul că sistemul s-a găsit la momentele $1, 2, \ldots, n-1, n$ respectiv în stările $i_1, i_2, \ldots, i_n$, nu depinde decît de ultima stare, adică

$$P(x_{n+1} = i_{n+1}/x_n = i_n,\ x_{n-1} = i_{n-1},\ \ldots,\ x_1 = i_1) = \\ = P(x_{n+1} = i_{n+1}\ /x_n = i_n).$$

Probabilitatea ca sistemul să fie în starea i la momentul n, o vom nota cu

$$P_i(n) = P(x_n = i).$$

Dacă considerăm aceste probabilități pentru orice stare $i = 1, 2, \ldots, m$, atunci, acestea formează un vector-linie, numit vector de stare, ale cărui componente satisfac relațiile:

$$0 \leqslant P_i(n) \leqslant 1,$$

$$\sum_{i=1}^{m} P_i(n) = 1.$$

Evident că numărul componentelor vectorului de stare $P(n)$ este egal cu numărul stărilor sistemului. Acest vector, dat fiind relațiile pe care le satisfac componentele sale, îl numim „vector stochastic".

Particularizînd pe n, adică considerînd momentul inițial (zero), vectorul de stare corespunzător îl vom nota prin analogie astfel:

$P(0) = (P_1(0), P_2(0), \ldots, P_m(0))$.

Probabilitatea ca sistemul să treacă în starea j la momentul $(n+1)$, știind că în momentul precedent n s-a aflat în starea i, adică:

$P(x_{n+1} = j/x_n = i) = P_{ij}\ (i, j = 1,2, \ldots, m)$

poartă numele de probabilitate de trecere.

Cu probabilitățile de trecere putem forma o matrice pătrată T, pe care o vom numi „matricea probabilităților de tre-

Scanned with OKEN Scanner

cere" ale cărei linii sînt vectori stochastici și ca atare elementele lor verifică relațiile:

$$0 \leqslant P_{ij} \leqslant 1 \qquad (i,j = 1, \ldots, m)$$

$$\sum_{j=1}^{m} P_{ij} = 1.$$

Un lanț Markov este complet determinat dacă cunoaștem vectorul de stare inițială $P(0)$ și matricea probabilităților de trecere $T = (P_{ij})$, $(i, j = 1, \ldots, m)$.

Relația prin care determinăm probabilitățile de stare la momentul $(n + 1)$, cu ajutorul probabilităților de trecere și a vectorului de stare corespunzător momentului n este:

$$P_j(n + 1) = \sum_{i=1}^{m} P_i(n)P_{ij},\ j = 1,\ldots, m.$$

Într-adevăr, să considerăm un sistem care îl putem găsi în una dintre stările $E_1, E_2, \ldots, E_8$ și fie n și $(n + 1)$ două momente consecutive.

Probabilitățile de stare în momentul n și cele pentru momentul $(n + 1)$ sînt date în figura IV.1.1. unde am figurat numai trecerile de la stările E_i $(i = 1, \ldots, 8)$ din momentul n la starea E_4 în momentul $(n + 1)$, cu scopul de a nu încărca prea mult figura.

Din desen rezultă că sistemul poate ajunge la momentul $(n + 1)$ în stare E_4, plecînd din oricare din stările E_1, $E_2, \ldots, E_8$, la momentul precedent n.

Dacă notăm cu $E_i^{(n+1)}$ $(i = 1,2, \ldots, 8)$ evenimentul ca sistemul să se găsească în starea E_i la momentul $(n + 1)$, cu $E_i^{(n)}$ $(i = 1,2, \ldots, 8)$ evenimentul ca sistemul să se găsească în starea E_i la momentul n, cu E_{i4}, evenimentul ca sistemul să treacă din starea E_i la momentul n în starea E_4 în momentul $(n + 1)$, avem următoarea relație:

$$E_4^{(n+1)} = (E_1^{(n)} \cap E_{14}) \cup (E_2^{(n)} \cup E_{24}) \cup \ldots \cup (E_8^{(n)} \cup E_{84}).$$

Ținînd seama că evenimentele $(E_i^{(n)} \cap E_{i4})$ i $= 1, 2, \ldots, 8$ sînt incompatibile, avem:

$$P(E_4^{(n+1)}) = P(E_1^{(n)} \cup E_{14}) + P(E_2^{(n)} \cup E_{24}) + \ldots + P(E_8^{(n)} \cap \cup E_{84}) = \sum_{i=1}^{8} P(E_i^{(n)} \cup E_{i4}).$$

Scanned with OKEN Scanner

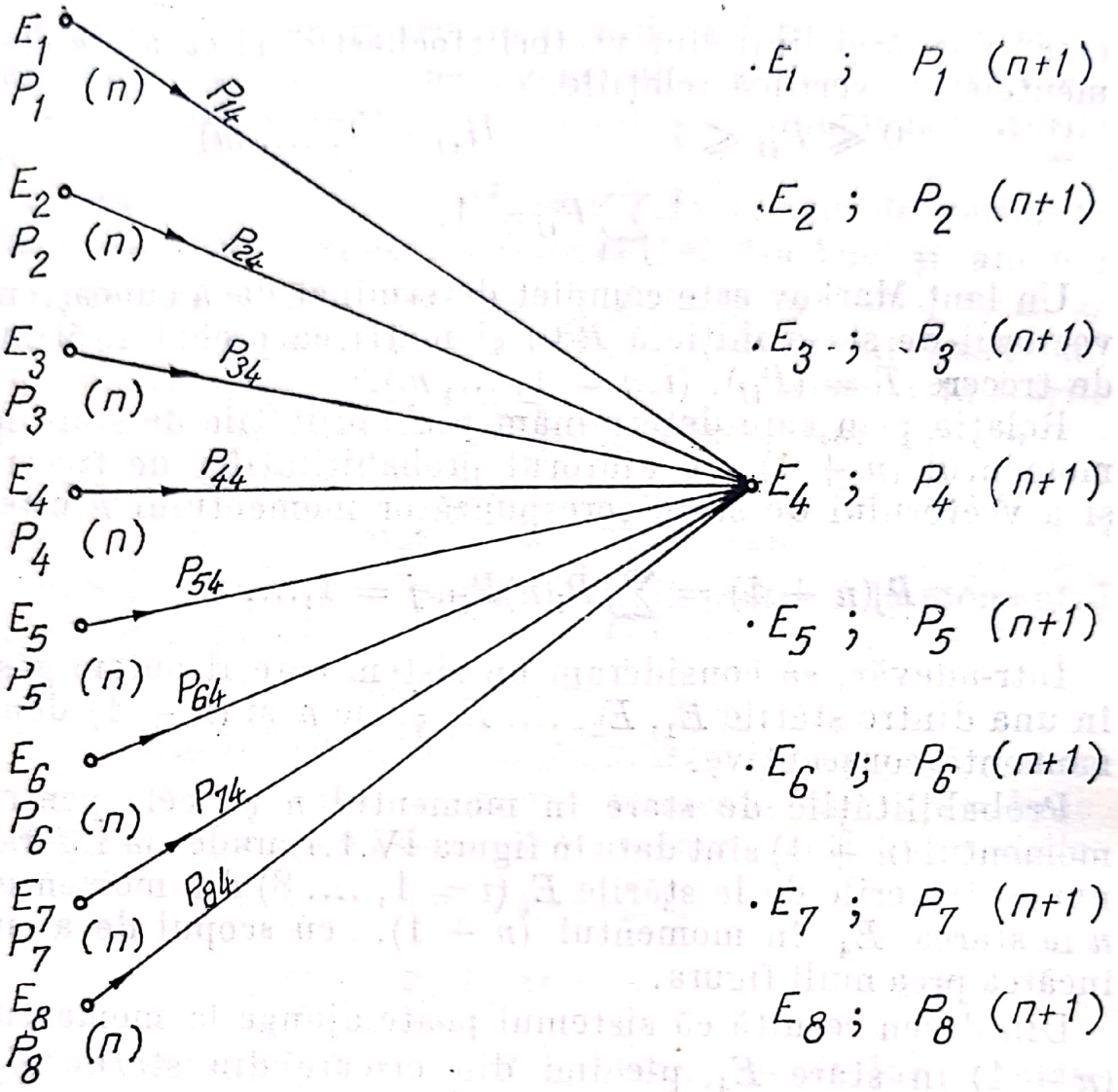


Fig. IV.1.1

Întrucît evenimentele $E_i^{(n)}$ și E_{i4} sînt independente, relația de mai sus devine:

$$P(E_4^{(n+1)}) = \sum_{i=1}^{8} P(E_i^{(n)}) \cdot P(E_{i4}).$$

Revenind la notațiile inițiale, avem:

$$P_4(n+1) = \sum_{i=1}^{8} P_i^{(n)} \cdot P_{i4}.$$

Scanned with OKEN Scanner

Starea E_4 la momentul $(n + 1)$ a fost arbitrar aleasă și deci sîntem îndreptățiți să scriem relația pentru cazul general, în care în loc de starea E_4 am avea o stare oarecare E_j:

$$P_j^{(n+1)} = \sum_{i=1}^{8} P_i^{(n)} \cdot P_{ij}.$$

sau încă, matriceal:

$$P(n + 1) = P(n) \cdot T, \qquad \text{(IV.1)}$$

unde T, este matricea probabilităților de trecere.

Menționăm că dacă probabilitățile de trecere nu depind de timp, atunci procesul este *staționar*.

Particularizînd relația (VI.1), obținem șirul de egalități:

$$P(n + 1) = P(n) \cdot T,$$
$$P(n + 2) = P(n + 1) \cdot T,$$
$$P(n + 3) = P(n + 2) \cdot T,$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$P(n + k - 1) = P(n + k - 2) \cdot T,$$
$$P(n + k) = P(n + k - 1) \cdot T.$$

Înmulțind membru cu membru egalitățile de mai sus și făcînd toate simplificările posibile, obținem relația:

$$P(n + k) = P(n) \cdot T(k). \qquad \text{(IV.2)}$$

În această relație, considerînd $n = 0$, obținem:

$$P(k) = P(0) \cdot T^{(k)}. \qquad \text{(IV.3)}$$

Elementele matricei $T^{(k)} = p_{ij}^{(k)} (i, j = 1, \ldots, m)$ reprezintă probabilitățile de trecere după k pași.

Egalitatea (IV.3) ne dă posibilitatea determinării matricei probabilităților de stare la momentul k, cunoscînd matricea probabilităților inițiale $P(0)$ și matricea probabilităților de trecere după k pași, $T^{(k)}$.

Scanned with OKEN Scanner

§ 2. COMPORTAREA ASIMPTOTICĂ A LANȚULUI MARKOV (ERGODICITATE)

În problemele cu caracter practic, de o deosebită importanță este punerea în evidență a unui regim limită sau regim permanent, în fenomenul stochastic studiat.

Am văzut că dacă se cunoaște legea de probabilitate a stării inițiale a sistemului, precum și matricea probabilităților de trecere, se poate determina legea de probabilitate pentru toate momentele ulterioare, lucru care a fost consemnat prin egalitatea

$$P(k) = P(0) \cdot T^{(k)}.$$

Egalitatea de mai sus precizează că starea viitoare a sistemului depinde de $P(0)$ și de matricea T a probabilităților de trecere, iar cunoașterea evoluției întregului proces se reduce la cunoașterea șirului de puteri succesive ale matricei T.

Sîntem conduși, pentru a cuncaște cît mai bine evoluția procesului în timp, la a determina limita:

$$\lim_{k\to\infty} T^{(k)} = \lim_{k\to\infty} \left(p_{ij}^{(k)}\right)_{i,j=1,2,\cdots,m} = \tau$$

sau

$$\tau = \lim_{k\to\infty} T^{(k)} = \lim_{k\to\infty} T^{(k)} \cdot T = \tau \cdot T.$$

Deci avem egalitatea:

$$\tau = \tau T, \tag{IV.4}$$

care ne permite găsirea elementelor matricei limită.

În cazul în care limita există, spunem că procesul Markov considerat este ergodic, sau că sistemul admite un regim permanent, sau, încă, sistemul admite o stare stabilă în probabilitate.

Dacă τ este independent de $P(0)$, lanțul Markov se zice că este regulat și matricea are toate liniile identice.

Scanned with OKEN Scanner

Fie, de exemplu, T matricea probabilităților de trecere,

$$T = \begin{Vmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{Vmatrix}$$

și matricea-limită

$$\tau = (x_1, x_2, x_3),$$

unde

$$0 \leqslant x_i \leqslant 1, \quad \sum_{i=1}^{3} x_i = 1.$$

Ne propunem să determinăm elementele matricei:

$$\tau T = (x_1, \ x_2, \ x_3) \begin{Vmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{Vmatrix} =$$

$$= \left(\frac{1}{2} x_1 + \frac{1}{2} x_2 + \frac{1}{4} x_3; \frac{1}{4} x_1 + \frac{1}{4} x_3; \frac{1}{4} x_1 + \frac{1}{2} x_2 + \frac{1}{2} x_3 \right).$$

Egalînd acest produs cu τ, în baza relației (IV.4) obținem sistemul:

$$\begin{cases} \frac{1}{2} x_1 + \frac{1}{2} x_2 + \frac{1}{4} x_3 = x_1, \\ \frac{1}{4} x_1 + \frac{1}{4} x_3 = x_2, \\ \frac{1}{4} x_1 + \frac{1}{2} x_2 + \frac{1}{2} x_3 = x_3, \end{cases}$$

Scanned with OKEN Scanner

Acesta este echivalent cu sistemul:

$$\begin{cases} \left(\frac{1}{2} - 1\right) x_1 + \frac{1}{2} x_2 + \frac{1}{4} x_3 = 0, \\ \frac{1}{4} x_1 + (0 - 1) x_2 + \frac{1}{4} x_3 = 0, \\ \frac{1}{4} x_1 + \frac{1}{2} x_2 + \left(\frac{1}{2} - 1\right) x_3 = 0. \end{cases}$$

Sistemul omogen de mai sus admite și soluții diferite de soluția banală, deoarece matricea corespunzătoare lui

$$\begin{vmatrix} \frac{1}{2} - 1 & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & 0 - 1 & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{2} - 1 \end{vmatrix} = T - E$$

are determinantul corespunzător nul. Într-adevăr, adunînd elementele tuturor liniilor la ultima, obținem o linie formată numai din zerouri.

Pentru ca sistemul nedeterminat, să devină unic determinat, atașăm acestuia și relația

$$\sum_{i=1}^{3} x_i = 1.$$

Avem în acest caz sistemul

$$\begin{cases} -\frac{1}{2} x_1 + \frac{1}{2} x_2 + \frac{1}{4} x_3 = 0, \\ \frac{1}{4} x_1 - x_2 + \frac{1}{4} x_3 = 0, \\ \frac{1}{4} x_1 + \frac{1}{2} x_2 - \frac{1}{2} x_3 = 0, \\ x_1 + x_2 + x_3 = 1. \end{cases}$$

Scanned with OKEN Scanner

sau încă,

$$\begin{cases} -\frac{1}{2}x_1 + \frac{1}{2}x_2 + \frac{1}{4}x_3 = 0, \\ \frac{1}{4}x_1 - x_2 + \frac{1}{4}x_3 = 0, \\ x_1 + x_2 + x_3 = 1. \end{cases}$$

Rezolvînd cu formulele lui Cramer obținem:

$$\Delta = \begin{vmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & -1 & \frac{1}{4} \\ 1 & 1 & 1 \end{vmatrix} = \frac{15}{16},$$

$$\Delta x_1 = \begin{vmatrix} 0 & \frac{1}{2} & \frac{1}{4} \\ 0 & -1 & \frac{1}{4} \\ 1 & 1 & 1 \end{vmatrix} = \frac{3}{8},$$

$$\Delta x_2 = \begin{vmatrix} -\frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{4} & 0 & \frac{1}{4} \\ 1 & 1 & 1 \end{vmatrix} = \frac{3}{16},$$

$$\Delta x_3 = \begin{vmatrix} -\frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{4} & -1 & 0 \\ 1 & 1 & 1 \end{vmatrix} = \frac{3}{8},$$

Scanned with OKEN Scanner

$$x_1 = \frac{\Delta x_1}{\Delta} = \frac{2}{5}$$

$$x_2 = \frac{\Delta x_2}{\Delta} = \frac{1}{5} \quad \text{şi deci} \quad \tau = \begin{vmatrix} 2/5 & 1/5 & 2/5 \\ 2/5 & 1/5 & 2/5 \\ 2/5 & 1/5 & 2/5 \end{vmatrix}.$$

$$x_3 = \frac{\Delta x_3}{\Delta} = \frac{2}{5}$$

§ 3. DETERMINAREA ELEMENTELOR MATRICEI-LIMITĂ CU AJUTORUL CALCULULUI MATRICIAL

Calculul elementelor matricei s-a făcut cu formulele lui Cramer. Dacă se ține seama însă de proprietățile specifice ale matricei T, calculele devin mai simple.

În acest scop, vom introduce mai întîi următoarele matrice auxiliare:

a) O matrice-linie cu n componente, notată cu L,

b) O matrice-coloană cu n componente, notată cu C, formate numai din elemente egale cu unitatea.

c) O matrice-unitate, notată cu E.

d) În matricea T a probabilităților de trecere, vom nota cu $T_{(r)}$ matricea formată din elementele situate pe linia r și cu $T^{(r)}$ matricea formată cu elementele coloanei r.

e) Matricea $M_r(T) = T - C \cdot T_{(r)}$

f) Matricea $M_r^*(T)$, care reprezintă minorul elementului p_{rr} ce se obține din matricea $M_r(T)$, suprimînd linia r și coloana r.

g) Matricea $T^*_{(r)}$ obținută din $T_{(r)}$ prin suprimarea elementului situat pe coloana r.

Cu ajutorul notațiilor introduse și ținînd seama de operațiile cu matrice, putem scrie:

1) $(A \cdot B)_{(r)} = A_{(r)} \cdot B$.

2) $C \cdot A_{(r)} = \begin{vmatrix} A_{(r)} \\ A_{(r)} \\ \cdots \\ A_{(r)} \end{vmatrix}$

Scanned with OKEN Scanner

Într-adevăr, acest produs dezvoltat îl scriem astfel:

$$C \cdot A_{(r)} = \begin{bmatrix} 1 \\ 1 \\ . \\ . \\ . \\ 1 \end{bmatrix}_{n\times 1} \cdot (a_{r1}\, a_{r2}, \ldots, a_{rn})_{1\cdot n} = \begin{bmatrix} a_{r1}\ a_{r2} \ldots, a_{rn} \\ a_{r1}\ a_{r2} \ldots\ldots a_{rn} \\ a_{r1}\ a_{r2} \ldots\ldots a_{rn} \\ \ldots\ldots\ldots\ldots\ldots \\ a_{r1}\ a_{r2} \ldots., a_{rn} \end{bmatrix}$$

adică o matrice cu toate liniile identice cu linia r din matricea A.

Întrucît matricea T este stochastică, avem următoarele relații:

a) $$T_{(r)}\, C = (a_{r1}\, a_{r2}, \ldots, a_{rn})_{1\times n} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ . \\ . \\ . \\ 1 \end{bmatrix}_{n\times 1} = a_{r1} + a_{r2} + \ldots + a_{rn} = 1.$$

b) $$T \cdot C = \begin{bmatrix} a_{11}\, a_{12}, \ldots, a_{1n} \\ a_{21}\, a_{22}, \ldots, a_{rn} \\ \ldots\ldots\ldots\ldots \\ a_{r1}\, a_{r2}, \ldots, a_{rn} \\ \ldots\ldots\ldots\ldots \\ a_{n1}\, a_{n}, \ldots, a_{nn} \end{bmatrix}_{n\times 1} \cdot \begin{bmatrix} 1 \\ 1 \\ . \\ . \\ . \\ 1 \\ . \\ . \\ . \\ 1 \end{bmatrix}_{n\times 1} =$$

Scanned with OKEN Scanner

$$= \begin{bmatrix} a_{11} + \ldots + a_{1n} \\ a_{21} + \ldots + a_{2n} \\ \\ a_{r1} + \ldots + a_{rn} \\ \\ a_{n1} + \ldots + a_{nn} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ . \\ . \\ . \\ 1 \end{bmatrix}_{n \cdot 1} = C.$$

Dacă avem satisfăcute simultan relațiile

$$T \cdot C = C \text{ și } L \cdot T = L, \tag{IV.5}$$

matricei T îi zicem *dublu stochastică*.

Din relația $\tau T = \tau$ rezultă

$$\tau \cdot (E - T) = 0. \tag{IV.6}$$

Elementele matricei τ fiind necunoscute, relația de mai sus este echivalentă cu un sistem omogen de n ecuații liniare, de forma $(x_1, x_2, .., x_n)(E - T) = 0$.
Dacă acestui sistem îi adăugăm și relația suplimentară

$$\sum_{i=1}^{n} x_i = 1,$$

sistemul devine:

$$\begin{cases} (x_1, x_2, \ldots, x_n)(E - T) = 0 \\ \sum_{i=1}^{n} x_i = 1. \end{cases} \tag{IV.7}$$

Sistemul (IV.7) are $n + 1$ ecuații și n necunoscute, însă primele ecuații nu sînt independente, deoarece determinantul

$$|E - T| = 0.$$

Dacă în prima relație din (IV.7) adăugăm în ambii membri produsul $(x_1, x_2, \ldots, x_n)\, C \cdot T_{(r)}$, obținem:

$$(x_1, \ldots, x_n)\,[E - (T - C \cdot T_{(r)})] = (x_1, \ldots, x_n) \cdot C \cdot T_{(r)} \tag{IV.8}$$

Scanned with OKEN Scanner

Știind că $\sum_{i=1}^{n} x_i = 1$ și că $(x_1, \ldots, x_n) \cdot C = 1$, iar $(x_1, \ldots, x_n)\ C \cdot T_{(r)} = T_{(r)}$, avem egalitatea:

$$(x_1, \ldots, x_n)\ [E - (T - C \cdot T_{(r)})] = T_{(r)}.$$

În baza definițiilor de mai sus obținem:

$$(x_1, \ldots, x_n)\ [E - M_r(T)] = T_{(r)}. \qquad \text{(IV.9)}$$

Orice matrice $(x_1, \ldots, x_n)$, care constituie o soluție a sistemului (IV.9), satisface condiția:

$$(x_1, \ldots, x_n) \cdot C = \sum_{i=1}^{n} x_i = 1.$$

Într-adevăr,

$$\sum_{i=1}^{n} x_i = (x_1, \ldots, x_n) \cdot C = (x_1, \ldots, x_n) \cdot [C - C + C] =$$

$$= (x_1, \ldots, x_n)\ [E - T + C \cdot T_{(r)}] \cdot C =$$

$$= (x_1, \ldots, x_n)\ [E - M_r(T)] \cdot C = T_{(r)} \cdot C = 1.$$

Pentru ca sistemul să aibă soluție unică este necesar ca determinantul

$$|E - M_r(T)| \neq 0$$

și $(x_1, \ldots, x_n) = T_{(r)}\ [E - M_r(A)]^{-1}$.

În modul acesta am demonstrat că dacă T este o matrice stochastică, atunci pentru orice $r = 1, 2, \ldots, n$ avem $E - M_r(T) \neq 0$, dacă și numai dacă sistemul relativ la T constă dintr-o singură clasă ergodică.

Exemplu. Fie matricea

$$T = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{bmatrix}$$

Scanned with OKEN Scanner

$$M_3(T) = T - \begin{bmatrix} T_{(3)} \\ T_{(3)} \\ T_{(3)} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{bmatrix} - \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & 0 & \frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$M_3^*(T) = \begin{bmatrix} \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{1}{4} \end{bmatrix},$$

unde M_3^* s-a obținut din $M_3(T)$, prin suprimarea liniei și coloanei a 3-a

$$E - M_3^*(T) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} \frac{1}{4} & 0 \\ \frac{1}{4} & -\frac{1}{4} \end{bmatrix} = \begin{bmatrix} \frac{3}{4} & 0 \\ -\frac{1}{4} & \frac{4}{4} \end{bmatrix}.$$

$$T_3^* = \left[\frac{1}{4}, \quad \frac{1}{4}\right],$$

unde $T_{(3)}^*$, s-a obținut suprimînd din linia $T_{(3)}$ elementul situat pe coloana a 3-a:

$$[E - M_3^*(T)]^{-1} = \frac{16}{15} \cdot \begin{bmatrix} 5/4 & 0 \\ 1/4 & 3/4 \end{bmatrix},$$

$$\lim_{n\to\infty} T_{(3)}^{*n} = T_{(3)}^* [E - M_{(3)}^*(T)]^{-1} = \frac{16}{15}\left(\frac{1}{4}, \frac{1}{4}\right) \cdot \begin{bmatrix} \frac{5}{4} & 0 \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix} =$$

$$= \frac{16}{15} \cdot \left(\frac{6}{16}, \frac{3}{16}\right) = \left(\frac{6}{15}, \frac{3}{15}\right) = \left(\frac{2}{5}, \frac{1}{5}\right);$$

cel de al treilea element se găsește știind că $\sum_i x_i = 1$. Deci,

$$\lim_{n\to\infty} T_{(3)}^n = \left(\frac{2}{5}, \frac{1}{5}, \frac{2}{5}\right)$$

Scanned with OKEN Scanner

și

$$\lim_{n\to\infty} T^n = \begin{bmatrix} 2/5 & 1/5 & 2/5 \\ 2/5 & 1/5 & 2/5 \\ 2/5 & 1/5 & 2/5 \end{bmatrix}.$$

§ 4. CORESPONDENȚA DINTRE LANȚURILE MARKOV ȘI GRAFE

1° **Introducere.** În continuare, ne propunem să facem o legătură între cele prezentate pînă acum și teoria grafelor.

Am văzut că unul din elementele caracteristice ale procesului stochastic de tip Markov este matricea probabilităților de trecere $T = (P_{ij})$, $i, j = 1,2, \ldots, m$.

Unei asemenea matrice i se poate asocia un graf ale cărui vîrfuri reprezintă mulțimea stărilor posibile ale procesului, legate prin arce care indică trecerea de la o stare la alta.

Aceasta permite ca anumite proprietăți ale lanțurilor Markov să fie studiate cu ajutorul teoriei grafelor.

Exemplul 1. Fie dată matricea probabilităților de trecere

$$T = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix};$$

graful corespunzător este redat în figura IV. 4.1.

Exemplul 2. Fie matricea probabilităților de trecere,

$$T = \begin{bmatrix} P_{11} & P_{12} & P_{13} & 0 & 0 & 0 & 0 \\ P_{21} & P_{22} & P_{23} & 0 & 0 & 0 & 0 \\ P_{31} & P_{32} & P_{33} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & P_{44} & P_{45} & 0 & 0 \\ 0 & 0 & 0 & P_{54} & P_{55} & 0 & 0 \\ 0 & 0 & 0 & P_{64} & P_{65} & P_{66} & P_{67} \\ 0 & 0 & 0 & P_{74} & P_{75} & P_{76} & P_{77} \end{bmatrix}$$

Scanned with OKEN Scanner

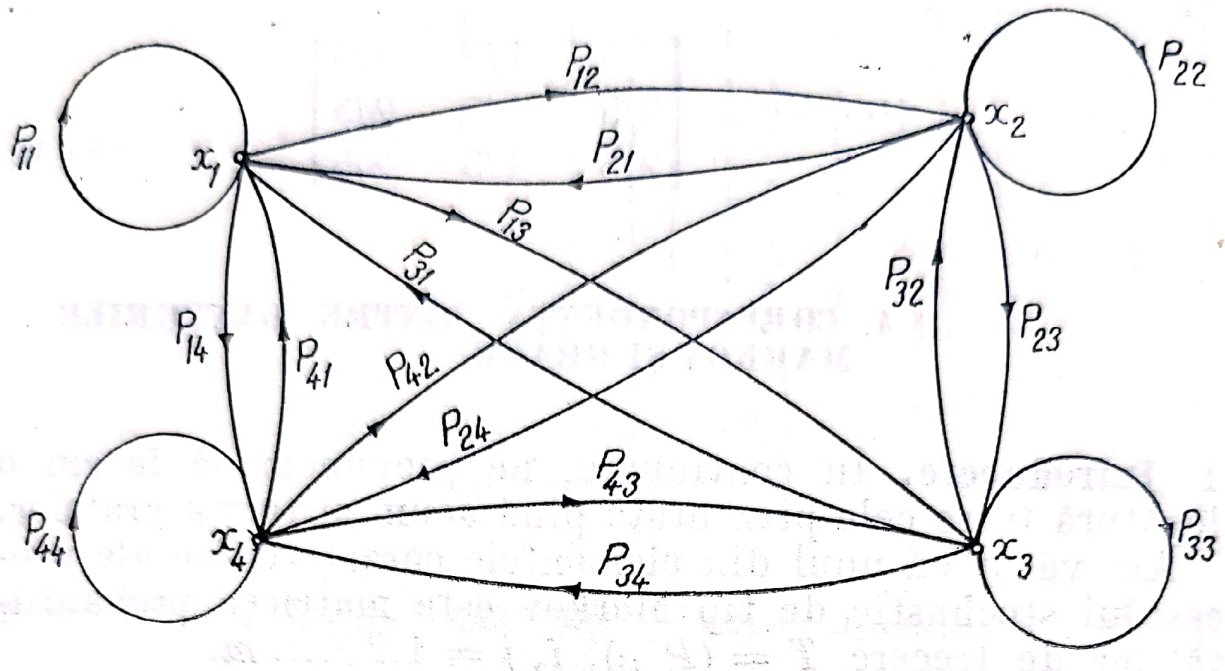


Fig. IV.4.1

Graful corespunzător este format din două componente distincte (figura IV.4.2).

Exemplul 3. Dacă considerăm drept matricea probabilităților de trecere, matricea

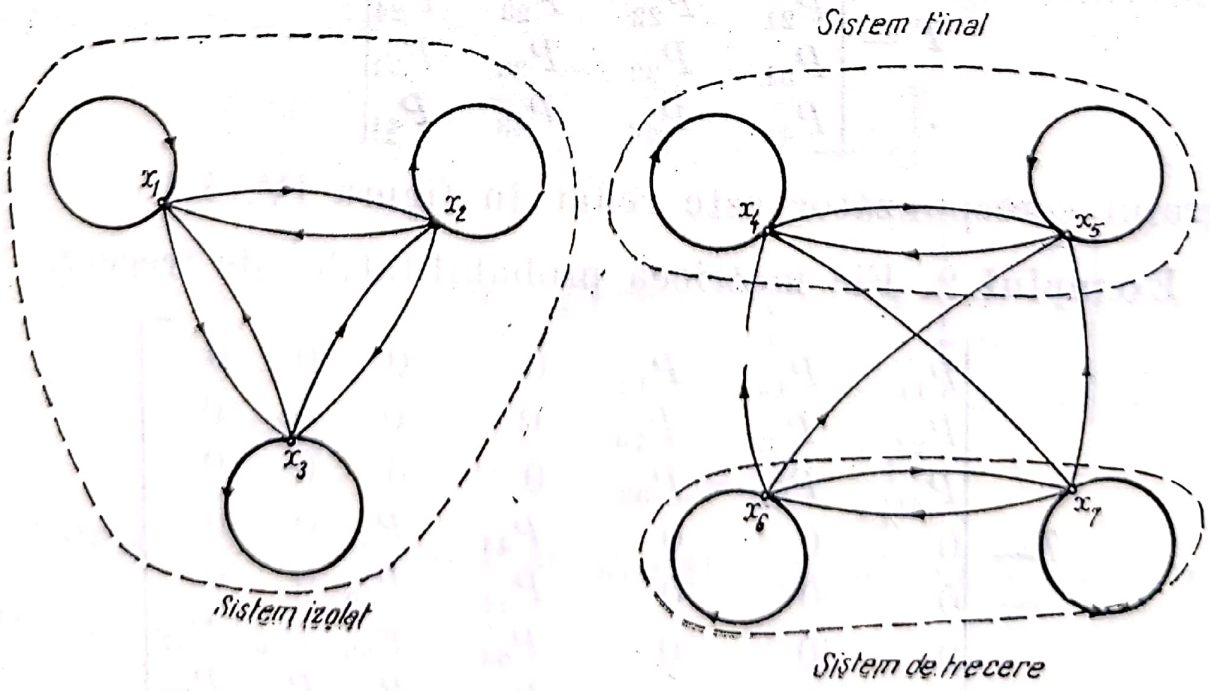


Fig. IV.4.2

Scanned with OKEN Scanner

$$T=\begin{vmatrix} 0 & p_{12} & p_{13} & 0 & p_{15} & p_{16} & p_{17} \\ 0 & P_{22} & P_{23} & 0 & 0 & 0 & 0 \\ 0 & P_{32} & P_{33} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & P_{44} & 0 & 0 & P_{47} \\ P_{51} & 0 & 0 & P_{54} & 0 & P_{56} & P_{57} \\ P_{61} & P_{62} & P_{63} & P_{64} & P_{65} & 0 & P_{67} \\ 0 & 0 & 0 & P_{74} & 0 & 0 & P_{77} \end{vmatrix}$$

avem graful din figura IV.4.3.

2° **Clasificarea stărilor unui lanț Markov cu ajutorul grafelor.** Posibilitatea asocierii procesului stochastic a unui graf în maniera arătată oferă posibilitatea clasificării stărilor acestuia. Mai întîi să dăm unele definiții.

Definiția 1. Numim matrice indecompozabilă matricea al cărei determinant nu îl putem descompune în produs de

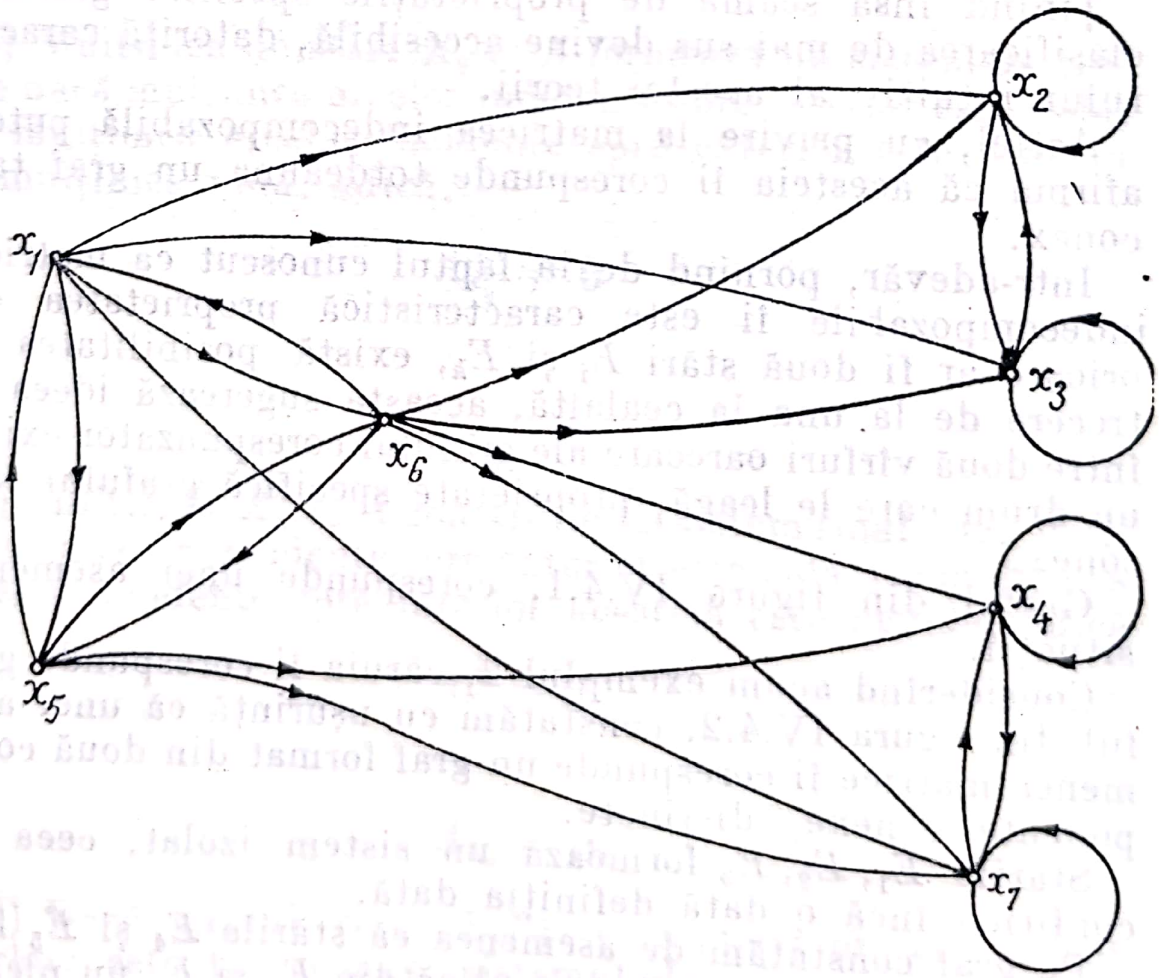


Fig. IV.4.3

Scanned with OKEN Scanner

determinanți; în caz contrar, matricea (și deci sistemul) este decompozabilă.

Definiția a 2-a. Numim sistem izolat mulțimea stărilor E_{i1}, E_{i2}, ..., E_{ik}, caracterizată prin aceea că probabilitățile de ieșire sau de intrare în acest sistem sînt nule.

Definiția a 3-a. Un sistem de stări E_{i1}, E_{i2}, ..., E_{ik} constituie un sistem final, dacă probabilitățile de intrare în acest sistem sînt diferite de zero, în timp ce probabilitățile de ieșire din sistem sînt nule.

Definiția a 4-a. Numim sistem de trecere o mulțime de stări pentru care probabilitatea de ieșire din sistem este diferită de zero, iar aceea de intrare este nulă.

Clasificarea stărilor unui lanț Markov, în lumina definițiilor de mai sus, în mod obișnuit o facem analitic, implicînd calcule greoaie.

Ținînd însă seama de proprietățile specifice grafelor, clasificarea de mai sus devine accesibilă, datorită caracterului intuitiv al acestei teorii.

Astfel, cu privire la matricea indecompozabilă putem afirma că acesteia îi corespunde totdeauna un graf tare conex.

Într-adevăr, pornind de la faptul cunoscut că matricei indecompozabile îi este caracteristică proprietatea că, oricare ar fi două stări E_i și E_k, există posibilitatea de trecere de la una la cealaltă, aceasta sugerează ideea că între două vîrfuri oarecare ale grafului corespunzător există un drum care le leagă, proprietate specifică grafului tare conex.

Graful din figura IV.4.1. corespunde unei asemenea situații.

Considerînd acum exemplul 2, căruia îi corespunde graful din figura IV.4.2, constatăm cu ușurință că unei asemenea matrice îi corespunde un graf format din două componente conexe, distincte.

Stările E_1, E_2, E_3 formează un sistem izolat, ceea ce confirmă încă o dată definiția dată.

Pe graf constatăm de asemenea că stările E_4 și E_5 formează un sistem final, în sensul că din E_4 și E_5 nu pleacă nici un arc spre exterior, în timp ce stările E_6 și E_7 for-

Scanned with OKEN Scanner

mează un sistem de trecere, adică nu au decît arce incidente spre exterior.

Graful din figura IV.4.3 admite două mulțimi finale distincte.

Sîntem acuma în măsură să facem o clasificare a stărilor unui lanț Markov, generalizînd rezultatele consemnate mai sus.

Fie $G = (X,F)$ graful corespunzător unui lanț Markov.

a) O mulțime de stări $X_1 \subset X$ formează un sistem „izolat", dacă atît mulțimea arcelor incidente spre exterior cît și mulțimea arcelor incidente spre interiorul acestei mulțimi sînt vide, adică

$$A^+_{x_1} = \emptyset$$

și

$$A^-_{x_1} = \emptyset$$

b) Mulțimea de stări $X_2 \subset X$ formează un sistem de trecere dacă mulțimea arcelor incidente spre interior este vidă, iar mulțimea arcelor incidente spre exterior este diferită de mulțimea vidă, adică,

$$A^+_{x_2} \neq \emptyset$$

și

$$A^-_{x_2} = \emptyset.$$

c) Mulțimea $X_3 \subset X$ constituie un sistem final dacă mulțimea arcelor incidente spre exterior este vidă, iar mulțimea arcelor incidente spre interior acesteia este nevidă, adică

$$A^+_{x_3} = \emptyset$$

și

$$A^-_{x_3} \neq \emptyset.$$

d) Dacă graful $G = (X,F)$ corespunzător unui lanț Markov este tare conex, sistemul de stări este indecompozabil.

Scanned with OKEN Scanner

§ 5. CONDIȚIA DE ERGODICITATE A UNUI LANȚ MARKOV, PREZENTATĂ CU AJUTORUL GRAFELOR

Toate cele prezentate pînă acum au avut drept scop să ofere posibilități de calcul pentru probabilitățile de trecere dintr-o stare oarecare E_i la momentul K în oricare altă stare E_j la momentul $(K + n)$, $n = 1,2, \ldots$

Deoarece, din punct de vedere economic, acestor treceri li se asociază anumite costuri (cîștiguri, pierderi, distanțe etc.), este clar că găsirea unei expresii simple a matricei trecerilor este foarte utilă.

Evident că dacă s-ar putea determina o matrice în care trecerile de la orice stare E_i la o stare E_j să se realizeze cu aceeași probabilitate independentă de starea inițială E_i, această situație ar implica calcule foarte simple.

Rezultă de aici necesitatea stabilirii apriorice a existenței sau a inexistenței regimului stabil, cu alte cuvinte, necesitatea de a stabili dacă lanțul este sau nu ergodic.

Metodele analitice folosite ne oferă acest răspuns după efectuarea tuturor calculelor corespunzătoare.

În cele ce urmează să arătăm cum teoria grafelor ne oferă posibilitatea determinării unor criterii de recunoaștere a ergodicității unui lanț Markov.

Am arătat în capitolul II importanța pe care o prezintă împărțirea vîrfurilor unui graf în clase de echivalență (subgrafe tare conexe). Această împărțire devine cu atît mai utilă, cînd graful studiat corespunde unui lanț Markov. Ținem să precizăm că în ceea ce privește studierea proceselor cu caracter aleator, interesează în mod deosebit acele clase de echivalență ale căror vîrfuri nu mai au urmași, numite și clase finale.

Astfel, dacă considerăm un graf oarecare $G = (X, F)$, descompus în clasele de echivalență ilustrate în figura IV.5.1, clasa finală corespunzătoare este formată din vîrfurile x_4 și x_8 care nu au urmași.

Pentru stabilirea unor proprietăți importante ale clasei finale, este necesar să amintim următoarele noțiuni ajutătoare.

Scanned with OKEN Scanner

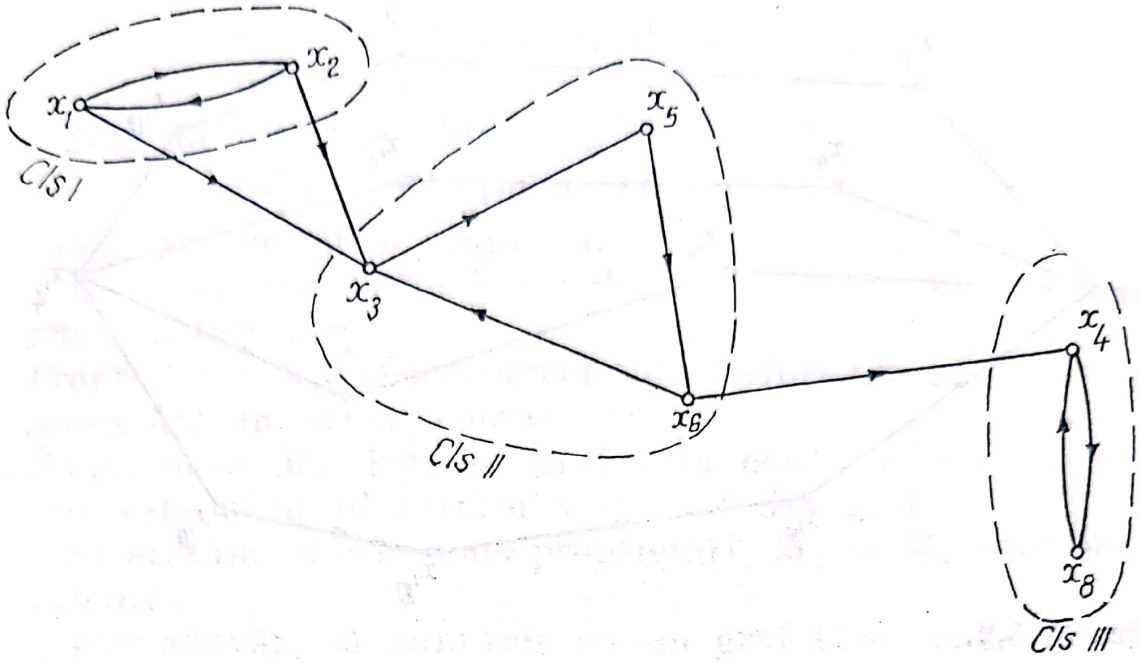


Fig. IV.5.1

Două numere x și y sînt egale modulo t dacă dau același rest prin împărțirea cu t.

Acest lucru se scrie:

$$x = y \pmod{t}$$

sau

$$|x - y| = 0 \pmod{t}.$$

De exemplu

$$128 = 7 \cdot 18 + 2,$$

$$58 = 7 \cdot 6 + 2,$$

adică 128 și 58 prin împărțirea cu 7 dau același rest 2, ceea ce ne permite să scriem:

$$128 = 58 \pmod{7}.$$

Pe de altă parte

$$128 - 58 = 70 = 0 \pmod{7}.$$

Dacă într-o clasă finală cel mai scurt circuit are lungimea t, atunci orice alt circuit al său are lungimea egală cu un multiplu de t (înțelegînd prin lungimea unui circuit numărul de arce pe care le conține).

Scanned with OKEN Scanner

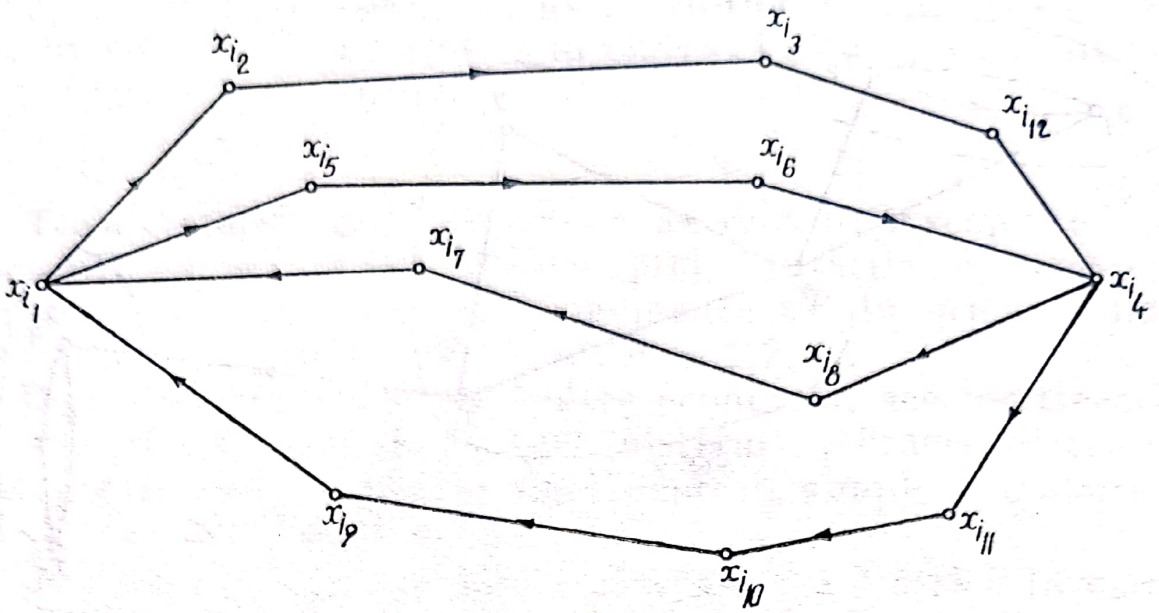


Fig. IV.5.2.

Într-adevăr, dacă considerăm drept clasă finală a unui graf $G = (X,F)$ graful din figura IV.5.2., lungimea minimă a circuitului este 7, și anume:

$$l(x_{i_1}, x_{i_5}, x_{i_6}, x_{i_8}, x_{i_7}, x_{i_1}) = 7.$$

Oricare alt circuit am considera, acesta va avea lungimea fie 7, fie un multiplu de 7.

$$l(x_{i_1}, x_{i_2}, x_{i_3}, x_{i_{12}}, x_{i_{14}}, x_{i_8}, x_{i_7}, x_{i_1}, x_{i_5}, x_{i_6}, x_{i_4}, x_{i_{11}}, x_{i_{10}}, x_{i_9}, x_{i_1}) = 14.$$

etc.

În general dacă i, j, i este un circuit, avem

$$l(i, j, i) = 0 \pmod 7.$$

Putem stabili acum următoarea proprietate importantă referitoare la lungimea drumului dintre două vîrfuri arbitrare ale clasei finale:

Oricare ar fi vîrfurile i și j și oricare ar fi drumurile ce le leagă, acestea au lungimile egale modulo t (unde t, așa cum am arătat, reprezintă lungimea minimă a circuitelor clasei finale).

Într-adevăr fie d_1 și d_2 două drumuri arbitrare între i și j și $l(d_1)$, $l(d_2)$ lungimile corespunzătoare. Dacă drumul d dintre i și j este astfel ales încît să formeze circuite cu drumurile d_1 și d_2, avem

$$l(d_1) + l(d) = 0 \pmod{t},$$
$$l(d_2) + l(d) = 0 \pmod{t}.$$

Scăzînd relațiile obținute, rezultă că

$$l(d_1) = l(d_2) \pmod{t},$$

adică proprietatea amintită.

În continuare, vom folosi două proprietăți evidente ale grafului tare conex.

Proprietatea M_1. Orice pereche de vîrfuri ale grafului tare conex are un urmaș comun.

Proprietatea M_2. Într-un graf tare conex există un vîrf care este urmașul tuturor vîrfurilor din graf.

Să arătăm că cele două proprietăți, M_1 și M_2, sînt echivalente.

Într-adevăr, să admitem că un graf tare conex posedă proprietatea M_2; în acest caz, el posedă și proprietatea M_1, deoarece toate vîrfurile grafului avînd un urmaș comun, cu atît mai mult oricare ar fi două dintre ele au urmaș comun, adică

$$M_2 \Rightarrow M_1.$$

Reciproc, dacă considerăm satisfăcută proprietatea M_1, să arătăm că este satisfăcută și proprietatea M_2.

Proprietatea M_1 afirmă că oricare ar fi perechea de vîrfuri x_i și x_j, acestea au un urmaș comun x_k.

Fie acum x_l un vîrf diferit de x_i și x_j. Vîrfurile x_l și x_k fiind două vîrfuri oarecare, au la rîndul lor un urmaș comun x_p, și ca atare vîrful x_p este urmașul comun atît al vîrfurilor x_i, x_j, cît și al vîrfului x_k. Dacă considerăm un alt vîrf x_r, diferit de toate celelalte considerate, există un urmaș comun al vîrfurilor x_p și x_r și ca atare, urmaș și pentru toate celelalte.

Continuînd raționamentul și ținînd seamă că graful este finit, ajungem la un vîrf x_f, urmaș al tuturor vîrfurilor grafului.

Prin urmare, existența proprietății M_1 implică existența proprietății M_2.

Implicațiile fiind reciproce,

$$\left.\begin{array}{l} M_2 \Rightarrow M_1 \\ M_1 \Rightarrow M_2 \end{array}\right\} \text{ rezultă } M_1 \equiv M_2,$$

ceea ce trebuia demonstrat.

Scanned with OKEN Scanner

Din echivalența proprietăților M_1 și M_2 deducem următoarele proprietăți:

a) Graful asociat unui lanț Markov care are proprietatea M_1 sau M_2 conține o singură clasă ergodică.

Într-adevăr, presupunem că graful admite mai multe clase ergodice, care în ultima analiză sînt niște subgrafe ale grafului considerat,

$$G_1, G_2, \ldots, G_p.$$

Să luăm două vîrfuri x_i și x_k aparținînd la două clase distincte, adică,

$$x_i \in G_i,$$
$$x_k \in G_k.$$

În baza proprietății M_1, această pereche de vîrfuri are un urmaș comun, de exemplu x_t, unde x_t aparține de asemenea uneia dintre clasele finale. Dat fiind faptul că fiecare clasă finală formează un graf tare conex, rezultă că vîrfurile x_i, x_t și x_k se găsesc pe același circuit.

Reiese că vîrful x_i este urmaș al vîrfului x_k și deci x_i aparține clasei G_k, pe de o parte, iar pe de altă parte, vîrful x_k este urmaș al vîrfului x_i și deci x_k aparține clasei G_i.

Rezultă deci că cele două clase finale G_i și G_k sînt formate din aceleași vîrfuri și ca atare nu sînt distincte. Continuînd raționamentul de mai sus din aproape în aproape, stabilim că cele p clase finale formează de fapt o clasă finală unică.

b) Dacă graful asociat unui lanț Markov are proprietatea M_1 sau M_2, atunci clasa sa ergodică nu se poate divide în subclase ciclice.

Să admitem prin absurd că această clasă ergodică se divide totuși într-un șir de subclase ciclice:

$$K_2, K_2, \ldots, K_r.$$

Fie x_i și x_j două vîrfuri care aparțin la două subclase ciclice distincte,

$$x_i \in K_i,$$
$$x_j \in K_j.$$

Conform proprietății M_1, cele două vîrfuri au un urmaș comun x_q, aparținînd de asemenea clasei ergodice și prin urmare unei subclase ciclice K_q, unde x_q va fi urmașul de ordinul $(q - i)$ mod.r al lui x_i și tot așa urmașul de

ordinul $(q - j)$ mod.r a lui x_j. Vîrful x_q fiind urmaș comun al vîrfurilor x_i și x_j, avem:

$$q - i \equiv q - j \pmod{r},$$

de unde

$$i \equiv j \pmod{r}$$

și ca atare, subclasele ciclice K_i și K_j sînt identice.

Pe baza aceluiași raționament, stabilim că toate subclasele ciclice $K_1, \ldots, K_r$ sînt identice.

Observații:

1. Deși am arătat echivalența celor două proprietăți M_1 și M_2 în justificarea proprietăților a și b, am folosit în exclusivitate proprietatea M_1, și aceasta întrucît cele două proprietăți sînt echivalente, pe de o parte, iar pe de altă parte, proprietatea M_1 este susceptibilă de un raționament mai simplu.

2. Proprietățile de mai sus, deși au fost stabilite numai pentru grafe asociate unor lanțuri Markov, ele rămîn adevărate, oricare ar fi graful $G = (X,F)$ tare conex.

Sîntem acum în măsură să dăm un criteriu cu ajutorul căruia să stabilim dacă un lanț Markov este sau nu ergodic, și anume:

Condiția necesară și suficientă ca un lanț Markov să fie ergodic, este ca graful asociat lui, să satisfacă proprietatea M_1 sau M_2.

Acest lucru rezultă, cu necesitate, din proprietățile a și b și observațiile 1 și 2.

O dată stabilită condiția necesară și suficientă ca un lanț Markov să fie ergodic, rămîne să stabilim mijloacele concrete de verificare a proprietăților M_1 sau M_2.

Pentru aceasta, să considerăm graful $G=(X,F)$ cu $|X|=n$, satisfăcînd proprietatea M_1.

În aceste condiții, vom arăta că oricare ar fi perechea de vîrfuri $x_i \in X$ și $x_j \in X$, există totdeauna un urmaș comun de ordinul S, unde

$$S \leqslant \frac{n(n-1)}{2}.$$

Scanned with OKEN Scanner

Într-adevăr, fiind satisfăcută condiția M_1, există un șir de perechi de vîrfuri de forma

$$(x_{i_0}, x_{j_0}); (x_{i_1}, x_{j_1}), \ldots, x_{i_0}, x_{j_0}),$$

astfel încît să avem:

a) $x_{i_k} \neq x_{j_k}$, vîrfuri din aceeași pereche, distincte,

b) $x_{i_k} \in F^k_{xi_0}$ și $x_{j_k} \in F^k_{x_{j_0}}$, adică vîrful x_{i_k} este urmașul de ordinul k al vîrfului inițial x_{i_0}, iar vîrful x_{j_k} este urmașul de ordinul k al vîrfului x_{j_0}.

c) $x_{i_\nu} = x_{j_\nu}$, adică ultima pereche o reducem de fapt la un simpul vîrf x_{i_ν}.

Dacă, analizînd șirul de perechi de vîrfuri, găsim două perechi identice, atunci drumul dintre ele va fi eliminat, reținînd numai cea de-a doua pereche.

Procedeul îl repetăm, pînă cînd obținem șirul:

$$(x_{i_0}, x_{j_0}); (x_{i_1}, x_{j_1}), \ldots, (x'_{i_p}, x'_{j_p}), \ldots, (x_{i_s}, x_{j_s}),$$

format numai din perechi distincte și satisfăcînd condițiile:

a′) $x_{i_p} \neq x_{j_p}$,

b′) $x_{i_p} \in F^p_{x_{i_0}}$ și $x_{j_p} \in F^p_{x_{j_0}}$

c′) $(x_{i_u}, x'_{j_u}) \neq (x_{i_v})$; $\begin{cases} u \neq v \\ u, v, = 1, 2, \ldots, s, \end{cases}$

d′) $x_{i_s} = x_{j_s}$.

Din b′ și d′ rezultă că $x_{i_s} = x_{j_s}$ este un urmaș de ordinul s al vîrfurilor (x_{i_0}, x_{j_0}), dar după a′ și c′, S este cel mult egal cu $\frac{n(n-1)}{2}$, deoarece cu cele n vîrfuri ale grafului putem forma cel mult $C^2_n = \frac{n(n-1)}{2}$ perechi distincte de vîrfuri.

Reamintim și de această dată că oricărui graf $G = (X, F)$ i se poate asocia o matrice booleană $B = (b_{ij})$, $i, j = 1, \ldots, n$, unde

$$b_{ij} = \begin{cases} 1 \text{ dacă există arc de la } x_i \text{ la } x_j, \\ 0 \text{ în caz contrar.} \end{cases}$$

Scanned with OKEN Scanner

De asemenea am arătat în capitolul II că puterea de ordinul k a matricei b_{ij} pune în evidență drumurile de lungime k între două vîrfuri arbitrare ale grafului; mai precis, dacă $b_{ij}^{(k)}$ este elementul generic al matricei B_k, avem

$$b_{ij}^{(k)} = \begin{cases} 1 \text{ dacă există un drum de lungime } k \text{ între } x_i \text{ și } x_j, \\ 0 \text{ în caz contrar.} \end{cases}$$

Combinînd acest ultim rezultat cu cel precedent, obținem o metodă practică de a verifica dacă un graf asociat unui lanț Markov satisface sau nu proprietatea M_1. În acest scop, calculăm puterile succesive ale matricei booleene asociate grafului lanțului dat. În momentul în care o anumită putere a acestei matrice are toate elementele egale cu unitatea, toate celelalte puteri mai mari decît k vor avea aceleași elemente, adică:

$$T_{(k+n)} = T_{(k)}, \text{ oricare ar fi } n > 1.$$

În această situație, proprietatea M_1 fiind satisfăcută, lanțul Markov este ergodic.

Dacă însă, începînd de la un anumit K, puterile consecutive ale matricei T coincid ($T^{(k+p)} = T^{(k)}$, $p>1$), deși T^k conține și elemente nule, proprietatea M_1 sau M_2 nu este satisfăcută și deci lanțul Markov considerat nu este ergodic.

Să reluăm exemplul tratat analitic în § 2, în care matricea probabilităților de trecere era:

$$T = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{bmatrix},$$

graful corespunzător fiind cel din figura IV.5.3.
Matricea booleană asociată grafului este:

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Scanned with OKEN Scanner

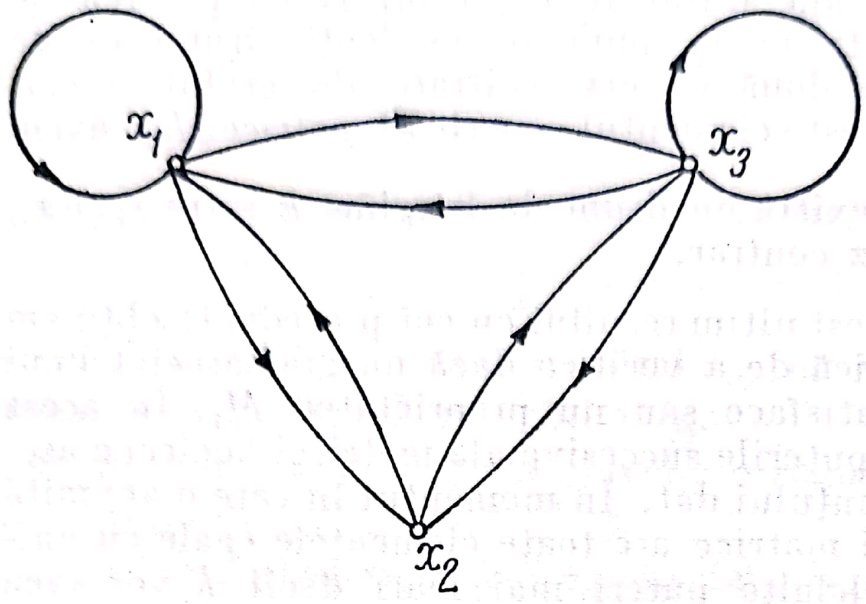


Fig. IV.5.3.

$$B^2 = B \cdot B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$B^3 = B^2.$$

În general, $B^k = B^2$, oricare ar fi $k > 2$; ca atare, în acest graf avem satisfăcute proprietățile M_1 și M_2, deci lanțul este ergodic, cum de altfel a rezultat și din calculele directe.

Să considerăm acum

Exemplul 2, unde matricea probabilităţilor de trecere este:

$$T = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{3} & \frac{1}{6} & 0 & 0 & \frac{1}{6} \end{bmatrix}$$

și graful corespunzător ei, cel din figura IV. 5.4.

Scanned with OKEN Scanner

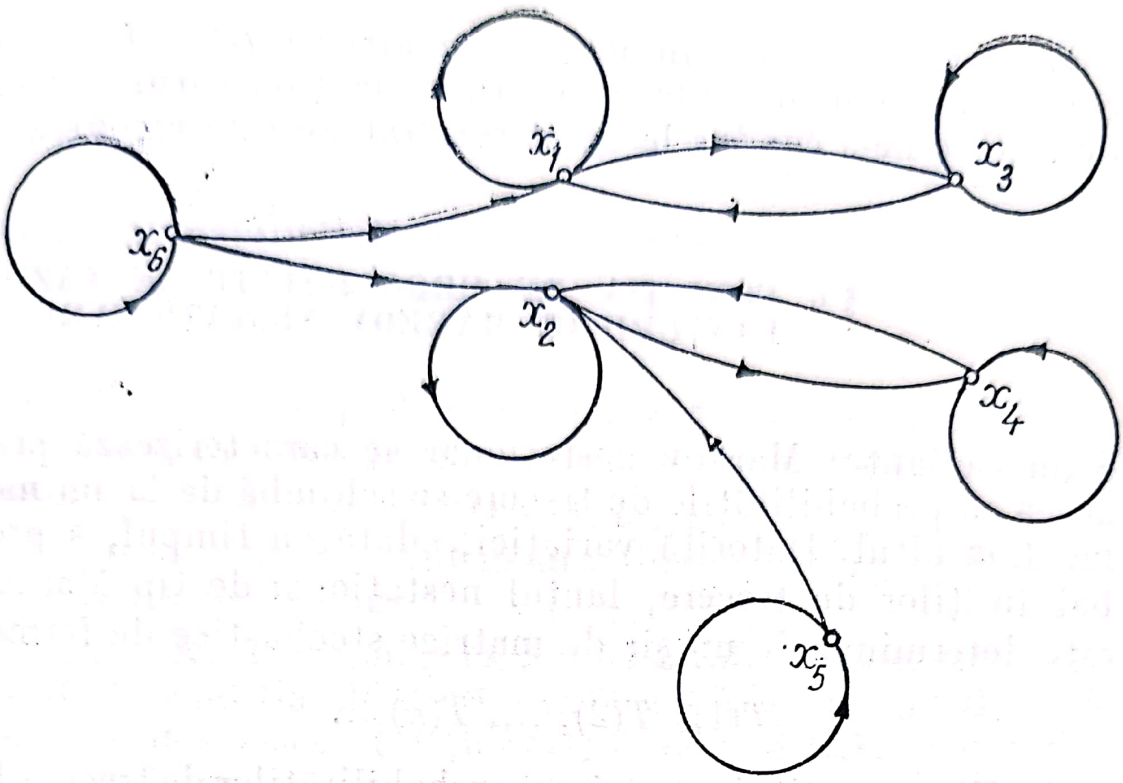


Fig. IV.5.4

Matricea booleană și puterile ei consecutive sînt cele de mai jos:

$$B = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$B^2 = B \cdot B = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$B^4 = B^2 \cdot B^2 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Scanned with OKEN Scanner

După cum se vede, în acest caz matricea $B^h = B^2 = B^4$ conține și elemente nule, prin urmare lanțul a cărui matrice de trecere ne-a condus la acest rezultat nu este ergodic.

§ 6. CONDIȚIA DE ERGODICITATE ÎN CAZUL LANȚURILOR MARKOV NESTAȚIONARE

Știm că lanțul Markov nestaționar se caracterizează prin aceea că probabilitățile de trecere se schimbă de la un moment la altul. Datorită variației, odată cu timpul, a probabilităților de trecere, lanțul nestaționar de tip Markov este determinat de un șir de matrice stochastice de forma:

$$T(1),\ T(2),\ \ldots,\ T(n),\ldots$$

unde $T(n)$ constituie matricea probabilităților de trecere la momentul n. Evident că numărul de matrice $T(n)$ poate să fie finit sau infinit.

În cazul lanțurilor Markov staționare, am arătat, cu ajutorul teoriei grafelor, cum putem cunoaște *apriori* dacă lanțul este sau nu ergodic. De această dată ne propunem să găsim condiții asemănătoare privitoare la ergodicitatea lanțurilor nestaționare.

Din teoria probabilităților știm că subzistă teorema: „Șirul $(P_{ij}^{(n)})_{n\in N}$ converge către o limită $(P_j^{\infty})_{j\in I}$, oricare ar fi $i \in I$, cînd n tinde către infinit, dacă și numai dacă există un $s > 0$ și o stare $h \in I$, astfel încît $P_{jh}^{(s)} > 0$ pentru orice $j \in I$, unde I constituie mulțimea stărilor procesului, considerată finită".

În legătură cu extensiunea la o mulțime arbitrară de stări ale lanțurilor Markov, amintim următoarea proprietate:

Alături de matricea probabilităților de trecere $(P_{ij})_{i,j\in I}$, să considerăm și funcția $P(x,A)$, unde A este o mulțime arbitrară de stări inclusă în I și pe care o definim astfel:

$$P(x,A) = \sum_{y\in A} p_{xy}.$$

Dacă considerăm și funcția

$$P^{(0)}(x, A) = \delta(x, A) = \delta(x, A) = \begin{cases} 1, & \text{dacă } x \in A, \\ 0, & \text{dacă } x \notin A, \end{cases}$$

atunci probabilitățile de trecere după n pași sînt:

$$P^{(n)}(x,A) = \sum_{y \in A} p_{xy}^{(n)}.$$

Studiul acestor probabilități, după cum vedem din relația de mai sus, îl reducem la studiul probabilităților de trecere după n pași, din matricea $(p_{ij})_{i,j \in I}$.

De asemenea, reamintim următoarea condiție pe care o vom folosi pe parcurs.

Condiția M. Vom spune că probabilitatea de trecere $P(x,A)$ verifică condiția M dacă există $s \in N$ și un $\mu > 0$, astfel încît pentru orice partiție $A_1 \cup A_2 = \emptyset$ $A_1 \cup A_2 = I$ a mulțimii I a stărilor să avem, $p^{(s)}(x,A_1) \geqslant \mu$ și $P^{(s)}(x,A_2) \geqslant \geqslant \mu$, sau

$$P^{(s)}(x,A_1) < 1 - \mu \quad \text{și} \quad P^{(s)}(x,A_2) < 1 - \mu.$$

Pe baza celor de mai sus, enunțăm teorema:

Pentru ca șirul $P^{(n)}(x,A)_{n \in N}$ să fie convergent către $P^{\infty}(A)$, este necesar și suficient să fie verificată proprietatea M.

Analizînd condițiile de ergodicitate a lanțurilor Markov nestaționare, deducem următoarele:

a) Oricărei matrice $T(n)$ din șirul $\{(T(n)\}_{n \in N}$ îi corespunde un graf finit, deoarece presupunem procesul n cu un număr finit de stări.

Condiția ca de la un anumit moment $s > 0$ să existe o stare $h \in I$ pentru care p_{jh} să fie strict pozitivă impune ca $p_{ij}^{(n)} > 0$ pentru $n \geqslant s$, oricare ar fi $i,j \in I$.

Sub aspectul teoriei grafelor, această condiție conduce la concluzia că, în cazul unui lanț Markov ergodic, graful corespunzător matricei probabilităților de trecere după m pași $(m \geqslant s)$ satisface condiția:

Oricare ar fi două vîrfuri x_i și x_j, acestea au un urmaș x_h comun (o aplicație comună).

În cele ce urmează, această condiție o vom nota cu M_3.

Întrucît lanțul Markov nestaționar este determinat de șirul de matrice stochastice $T(n)_{n\in N}$, matricea probabilităților de trecere după m pași o obținem făcînd produsul primelor m matrice din acest șir, adică:

$$P(m) = \prod_{i=1}^{m} T(i).$$

Pe baza celor de mai sus, putem formula o condiție de ergodicitate a lanțurilor Markov nestaționare, și anume:

Condiția necesară și suficientă ca un lanț Markov nestaționar să fie ergodic, este ca el să satisfacă proprietatea M_3.

Condiția este necesară, deoarece din ergodicitatea lanțului Markov am constatat că rezultă proprietatea M_3.

Condiția este suficientă. Pentru justificare, vom considera șirul de matrice $P(m)_{m=1,2,\ldots,n,\ldots}$, care presupunem că satisfac proprietatea M_3.

Mai întîi, să introducem următoarele notații:

$T(k)$, un element arbitrar din produsul $P(m)$, unde $T(k) = (p_{ij}^{(k)})_{i,j\in I}$,

$$t_j^{(k)} = \sup_i p_{ij}^{(k)},$$

$$t_j^{(k)} = \inf_i p_{ij}^{(k)},$$

$$\mu = \inf_{i,j,k} p_{ij}^{(k)},$$

Să considerăm acum două linii arbitrare a și b din matricea $T(k)$ și să formăm suma diferențelor elementelor corespunzătoare lor. Avem:

$$S_{a,b} = \sum_{k\in I} (p_{ak} - p_{bk}). \tag{1}$$

Aplicînd operatorul sumă asupra fiecărui termen al diferenței și știind că suma elementelor unei linii într-o matrice stochastică este unu, putem scrie:

$$S_{a,b} = \sum_{k\in I} p_{ak} - \sum_{k\in I} p_{bk} = 1 - 1 = 0. \tag{2}$$

Termenii sumei $S_{a,b}$, pot fi pozitivi, negativi, sau nuli, după cum

$$p_{ak} > p_{bk},\ p_{ak} < p_{bk},\ \text{sau}\ p_{ak} = p_{bk}.$$

Scanned with OKEN Scanner

Aceasta ne dă posibilitatea grupării termenilor sumei $S_{a,b}$ în două sume distincte $S^+_{a,b}$, care să conţină toate diferenţele pozitive sau nule, şi $S^-_{a,b}$, care conţine toţi termenii negativi.

Vom nota cu I^+ indicii K pentru care diferenţele sînt pozitive sau nule şi cu I^- ceilalţi indici. În acest fel, mulţimea stărilor procesului poate fi parţionată corespunzător în două submulţimi,

$A_1 \subset I^+$ şi $A_2 \subset I^-$, pentru care $A_1 \cap A_2 = \emptyset$ şi $A_1 \cup A_2 = I$.

Vom avea:

$$S_{a,b} = \sum_{p_{ak} \geqslant p_{bk}} (p_{ak} - p_{bk}) + \sum_{p_{ak} < p_{bk}} (p_{ak} - p_{bk})$$

sau

$$S_{a,b} = S^+_{a,b} + S^-_{a,b}.$$

Din (2) rezultă că:

$$S^+_{a,b} = |\, S^-_{a,b} \,|\,.$$

Am presupus de la bun început însă că grafele asociate matricelor $\{T^{(n)}\}_{n \in N}$ satisfac M_3. Ca atare, vîrfurile x_a şi x_b au o aplicaţie comună, x_{k^*} şi deci există probabilităţile strict pozitive p_{ak^*} şi p_{bk^*}.

Diferenţa $(p_{ak^*} - p_{bk^*})$ poate fi la rîndul ei pozitivă, negativă, sau nulă şi prin urmare poate să aparţină fie lui S^+_{ab}, fie lui $S^-_{a,b}$.

Să analizăm fiecare din aceste două situaţii:

1°. $\qquad (p_{ha}{}^+ - p_{bk}{}^+) \in S^+_{a,b}.$

Putem scrie:

$$S^+_{a,b} = \sum_{p_{ak} \geqslant p_{bk}} (p_{ak} - p_{bk}) = \sum_{p_{ak} \geqslant p_{bk}} p_{ak} - \sum_{p_{ak} \geqslant p_{bk}} p_{bk} \leqslant$$

$$\leqslant \sum_{k \in i} p_{ak} - p_{bk^*} = 1 - p_{bk^*} \leqslant 1 - \mu,$$

deoarece $p_{bk^*} > \mu$.

Scanned with OKEN Scanner

Analog se arată că:

$$|S_{a,b}^{-}| = \Big|\sum_{p_{ak}<p_{bk}} (p_{ak} - p_{bk})\Big| \leqslant 1 - \mu.$$

Să considerăm în continuare produsul a două matrice $T(k)$ și $T(q)$ din expresia produsului $P(m)$ satisfăcînd M_3. Cu privire la matricea $T(q)$ și matricea produs $T(k,q) = T(k) \cdot T(q)$, introducem notațiile

$$\bar{t}_j^{(q)} = \sup_i p_{ij}^{(q)},$$

$$\underline{t}_j^{(q)} = \inf_i p_{ij}^{(q)},$$

$$\bar{t}_j^{(kq)} = \sup_i p_{ij}^{(kq)},$$

$$\underline{t}_j^{(kq)} = \inf_i p_{ij}^{(kq)},$$

Elementul $p_{ij}^{(kq)}$ al matricei produs îl obținem după regula de înmulțire a matricelor și anume:

$$p_{ij}^{(k,q)} = \sum_{r \in I} p_{ir}^{(k)} \, p_{rj}^{(q)} \; ; \; i,j, \in I.$$

Fie acum în matricea produs $T(k,q)$ două linii oarecare c și d. Un element de pe linia c, respectiv d, din matricea produs este dat de relațiile:

$$p_{cr}^{(k,q)} = \sum_{e \in I} p_{ce}^{(k)} \, p_{er}^{(q)},$$

$$p_{dr}^{(kq)} = \sum_{e \in I} p_{de}^{(k)} \, p_{er}^{(q)}.$$

Făcînd diferența dintre elementele corespunzătoare acestor linii obținem:

$$p_{cr}^{(kq)} - p_{dr}^{(kq)} = \sum_{e \in I} p_{ce}^{(k)} \, p_{er}^{q()} - \sum_{e \in I} p_{de}^{(k)} \, p_{er}^{(q)} =$$

$$= \sum_{e \in I} (p_{ce}^{(k)} - p_{de}^{(k)}) \, p_{er}^{(q)}$$

și ținînd seama de notațiile introduse avem:

$$\bar{t}_r^{(kq)} - \underline{t}_r^{(kq)} = \sup_{c,d} (p_{cr}^{(kq)} - p_{dr}^{(kq)}) = \sup_{c,d} \sum_{i \in I} (p_{ce}^{(k)} - p_{de}^{(k)}) \times$$

$$\times p_{er}^{(q)} = \sup_{c,d} \sum_{\geqslant p_{de}} (p_{ce}^{(k)} - p_{de}^{(k)}) p_{er}^{(q)} + \sup_{c,d} \sum_{p_{ce} < p_{de}} (p_{ce}^{(k)} -$$

$$- p_{de}^{(k)}) p_{er}^{(k)} \leqslant (1 - \mu)\bar{t}_r^{(q)} - (1 - \mu)\underline{t}_r^{(q)} = (1 - \mu)(\bar{t}_r^{(q)} -$$

$$- \underline{t}_r^{(q)}) \leqslant (1 - \mu)^2$$

Din aproape în aproape, arătăm că pentru produsul a m matrice din șirul $T(n)_{n \in N}$, avem relația:

$$\bar{t}_j^{P(m)} - \underline{t}_j^{P(m)} \leqslant (1 - \mu)^m.$$

Din ultima relație rezultă ergodicitatea lanțurilor Markov nestaționare, deci suficiența condiției M_3.

§ 7. LEGĂTURA GRAFELOR CU TEORIA JOCURILOR

Știm că teoria jocurilor poate fi definită ca o teorie matematică a situațiilor de conflict. Astfel de situații conflictuale apar în multiple domenii de activitate, printre care și în cele cu caracter economic. Pentru a putea studia asemenea situații cu ajutorul aparatului matematic, ele trebuie mai întîi încadrate într-un model care permite deducerea unor concluzii asupra modului în care trebuie rezolvat conflictul.

Într-un joc apar de obicei părțile în conflict, care trebuie să întocmească un plan de acțiune de așa manieră, încît unul sau celălalt dintre parteneri să cîștige confruntarea.

Evident că toată dibăcia cîștigătorului constă în felul cum va întreprinde acțiunile de la o etapă la alta, acțiuni impuse și de adversar. De aici și noțiunea de „strategie" a jocului, prin care vom înțelege arta de a folosi cu rațiune mulțimea mijloacelor disponibile în vederea succesului final. Menționăm că dacă un jucător are la dispoziție în desfă-

Scanned with OKEN Scanner

şurarea completă a unui joc m alternative, spunem că partenerul dispune de m strategii pure. Cînd confruntările (partidele) se repetă, adversarii au posibilitatea de a alege strategiile pure cu anumite frecvenţe, şi, în acest caz, vom spune că ei au folosit strategii mixte.

Caracteristică unui joc strategic îi este şi noţiunea de „cîştig", înţelegînd prin aceasta rezultatul confruntării a două strategii pure alese de jucători. Aceste cîştiguri sînt diferite, deoarece obiectivele lor sînt diferite, adversarii urmărind realizarea aceluiaşi eveniment.

Fiind dat un joc, avem posibilitatea să-i asociem acestuia unui graf arborescent care ne dă o imagine intuitivă a ordinii mutărilor şi a tuturor informaţiilor, începînd cu prima etapă şi terminînd cu ultima. Fiecare vîrf al grafului va reprezenta poziţia jucătorului, iar muchiile sale arată posibilităţile pe care le are jucătorul de la o mutare la alta.

Exemplu. Pe graful din figura IV.7.1 putem urmări toate posibilităţile pe care le poate adopta fiecare din jucătorii x_1 şi x_2.

Jucătorul x_1 poate adopta mişcările distincte (strategii pure)

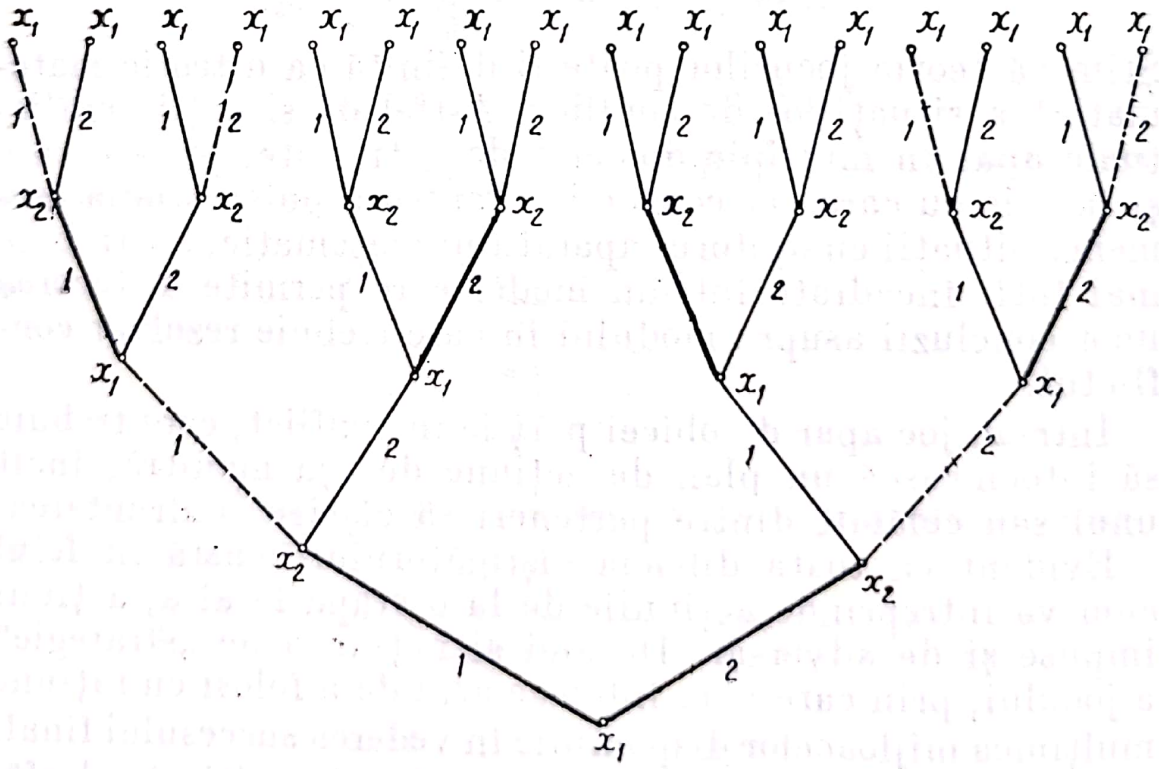


Fig. IV.7.1.

Scanned with OKEN Scanner

(1,1); (1,2); (2,1); 2,2) marcate prin linie îngroșată. Jucătorul x_2, are la dispoziție tot patru strategii pure (1,1); (1,2); (2,1); (2,2), marcate prin linie întreruptă.

Se cunoaște de asemenea că un joc în forma normală se prezintă sub forma unei matrice care pune în evidență strategiile pure ale jucătorului x_1, strategiile pure ale jucătorului x_2 și cîștigurile C_{ij}, realizate de x_1, în ipoteza că el a ales strategia i în timp ce x_2 a ales strategia j. Schimbînd semnele elementelor acestei matrice obținem matricea cîștigurilor realizate de jucătorul x_2.

Strategia pură a lui x_1 \ Strategia pură a lui x_2	$x_2^{(1)}$ $x_2^{(2)}$ . . . . . $x_2^{(1)}$ . . . . . $x_2^{(n)}$
$x_1^{(1)}$	C_{11} C_{12} . . . . C_{1j} . . . . C_{1n}
$x_1^{(2)}$	C_{21} C_{22} . . . . C_{2j} . . . . C_{2n}
.	
.	
.	
.	
.	
$x_1^{(i)}$	C_{i1} C_{i2} . . . . C_{ij} . . . . C_{in}
.	
.	
.	
.	
.	
$x_1^{(n)}$	C_{n1} C_{n2} . . . . C_{nj} . . . . C_{nn}

Cu ajutorul matricei cîștigurilor putem, de asemenea, construi un graf în care vîrfurile sale reprezintă strategiile pure ale jucătorilor, iar existența arcului (x_i, x_j) în graf implică alegerea strategiei x_i, de către primul jucător și a strategiei x_j, de către cel de-al doilea jucător, iar capacitățile arcelor (x_i, x_j) reprezintă cîștigul realizat de primul jucător în această ipoteză.

Scanned with OKEN Scanner

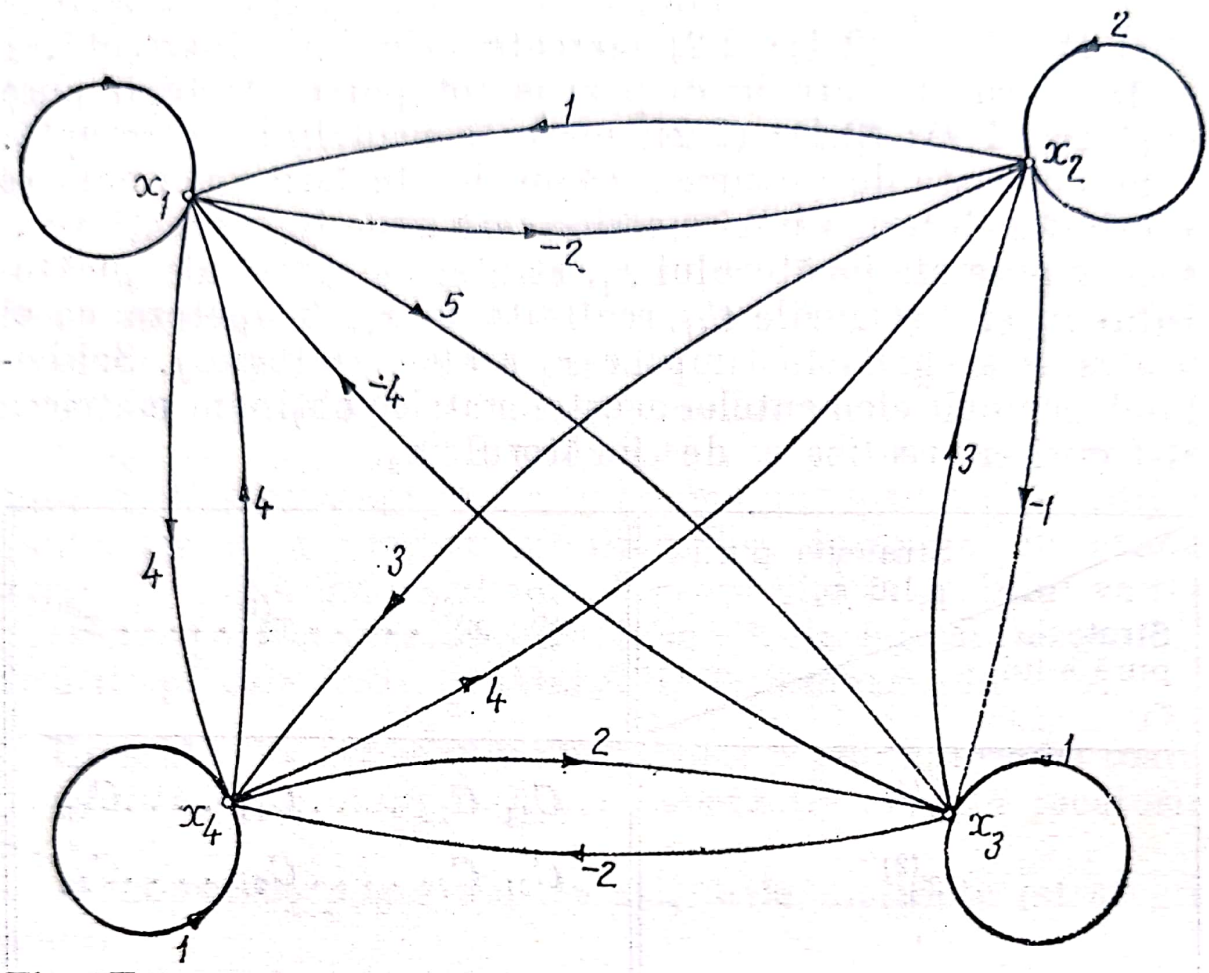


Fig. IV.7.2.

Admitem că matricea cîștigului este:

	x_1	x_2	x_3	x_4
x_1	3	—2	5	4
x_2	1	2	—1	3
x_3	—4	3	1	—2
x_4	4	4	2	1

Fiind dat un graf $G = (X,F)$ sau $G = (X,A)$, putem defini un joc pe el dacă ținem seama de regulile de mai jos.

Stabilim printr-o convenție jucătorul care își alege un vîrf oarecare al grafului, x_{i_1}.

Jucătorul al doilea va fi obligat să-și aleagă un vîrf din mulțimea aplicațiilor lui x_{i_1}, adică Fx_{i_1}. Fie x_{i_2} vîrful preferat de-al doilea jucător, atunci primul jucător evident

Scanned with OKEN Scanner

va alege un vîrf din mulțimea aplicațiilor lui x_{i_2} (Fx_{i_2}) și tot așa mai departe, pînă cînd ajungem la un vîrf x_{i_p} pentru care mulțimea aplicațiilor sale este vidă,

$$Fx_{i_p} = \emptyset.$$

Pentru o asemenea confruntare, apare evidentă necesitatea alegerii strategiilor jocului de așa manieră, încît aceasta să ducă la cîștigarea partidei, adică la alegerea vîrfului x_p, pentru care

$$Fx_p = \emptyset.$$

În capitolul I am arătat ce este acela nucleul grafului și cum se poate el determina. În teoria jocurilor, nucleul grafului este de o mare importanță, deoarece cu ajutorul lui putem stabili strategia pe care trebuie să o adopte un jucător pentru ca el să fie cîștigător.

Într-adevăr, se știe că nucleul este format dintr-o mulțime de vîrfuri interior și exterior stabile sau, încă, mulțimea vîrfurilor nucleului se bucură de proprietățile:

a) Nu există arce care să le lege, adică aplicația unui vîrf al nucleului nu aparține nucleului.

b) Oricărui vîrf dinafara nucleului îi corespunde o aplicație în nucleu.

Să admitem că graful $G = (X,F)$ admite nucleul

$$N = \{ x_{i_1}, x_{i_2}, \ldots, x_{i_j}, \ldots, x_{i_k} \}.$$

Admitem că la un moment dat primul jucător alege un vîrf $x_{i_j}^{(1)}$ aparținînd nucleului. În baza celor arătate mai sus, $Fx_{i_j}^{(1)} \cap N = \emptyset$; deci jucătorul J_2 este obligat să aleagă un vîrf $x_p^{(1)}$ dinafara nucleului.

Întrucît $Fx_p^{(1)} \cap N \neq \emptyset$, jucătorul J_1 poate alege din nou un vîrf $x_{i_j}^{(2)}$ din același nucleu, jucătorul J_2 fiind din nou obligat să aleagă un vîrf dinafara nucleului. Jocul continuă, pînă cînd în mod obligatoriu jucătorul J_1 va alege un vîrf $x_{i_j}^{(r)}$ din nucleu pentru care $Fx_{i_j} = \emptyset$ și deci el este cîștigătorul jocului.

Să exemplificăm cele de mai sus pe graful din figura IV.7.3, cu

$$N = \{ x_1, x_3, x_5 \}.$$

Scanned with OKEN Scanner

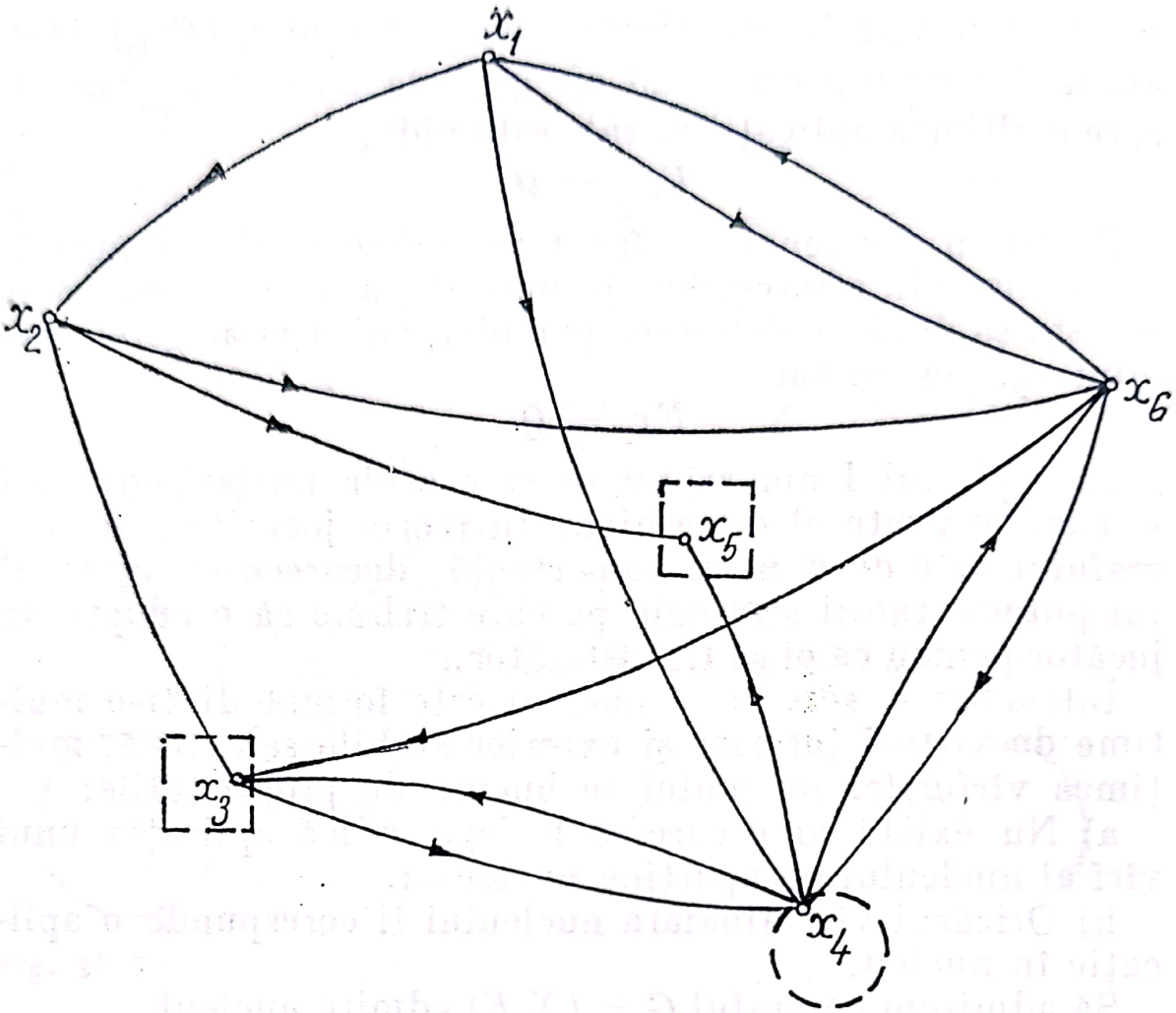


Fig. IV.7.3.

Presupunem că jucătorul J_1 a ales vîrful x_3 marcat pe graf. Constatăm că J_2 va fi obligat să aleagă unul din celelalte vîrfuri x_2, x_4, x_6.

Fie x_4 vîrful ales de jucătorul J_2. Jucătorul J_1 va alege vîrful x_5 din nucleu, deși are și posibilitatea alegerii vîrfului x_6.

Mulțimea Fx_5 fiind vidă, jocul s-a terminat în favoarea jucătorului J_1.

Observație: Jucătorul care alege un vîrf al nucleului va cîștiga partida numai dacă va urmări tot timpul să se mențină la alegerea vîrfurilor aparținînd nucleului. În cazul în care ar părăsi alegerea unui vîrf al nucleului, ar da posibilitatea ca adversarul său să intre în alegerea vîrfurilor nucleului, putînd pierde în acest fel partida.

Scanned with OKEN Scanner

ANEXĂ

§ 1. NOȚIUNI DE TEORIA MULȚIMILOR

1°. Conceptul de mulțime fiind considerat ca o noțiune primară nu îl definim; el se poate clarifica prin exemple. Astfel, vom spune că orice colecție de obiecte în număr finit sau infinit, având anumite caracteristici comune, formează o „mulțime".

Așa de exemplu, putem vorbi de mulțimea tuturor numerelor naturale, de mulțimea numerelor reale, de mulțimea tuturor cooperativelor de producție dintr-un județ ș.a.

Obiectele din care este alcătuită o mulțime le numim „elementele" mulțimii și ele pot fi de orice natură, fie concrete, fie obiecte ale gîndirii.

Mulțimile și elementele le notăm prin litere; de obicei, mulțimea o notăm cu literă mare, iar elementul cu litere mici.

O mulțime o considerăm cunoscută dacă se dau elementele sale sau o proprietate comună tuturor elementelor mulțimii, spre deosebire de elementele unei alte mulțimi.

Exemple

a) Mulțimea formată din numerele $2,4,5$, $\sqrt{31}$, π o vom scrie $A = \{ 2,4,5, \sqrt{31}, \pi \}$, incluzînd elementele sale $2,4,5$, $\sqrt{31}$, π între acolade.

b) Mulțimea numerelor naturale pe care o notăm N este

$$N = \{ 1,2,3,\ldots,n,\ldots \}.$$

c) Mulțimea numerelor reale cuprinse între 2 și 5 se scrie

$$A = \{ x \mid 2 < x < 5 \},$$

Scanned with OKEN Scanner

adică această mulțime este formată din numerele x cu proprietatea că sînt cuprinse între 2 și 5 (această proprietate am specificat-o după liniuța verticală).

d) Mulțimea formată cu rădăcinile polinomului

$$P(x) = 3x^5 + x^3 - x + 2$$

o vom scrie

$$A = \{ x \mid 3x^5 + x^3 - x + 2 = 0 \}.$$

Dacă ne interesează însă numai rădăcinile reale ale acestui polinom, vom completa proprietatea de mai sus și prin precizarea că x trebuie să fie un număr real.

Potrivit celor arătate pînă acum vom considera și mulțimea formată dintr-un singur element a, de pildă. Atunci trebuie făcută distincția între mulțimea (a) și elementul a.

2°. **Relația de apartenență și incluziune.** Faptul prin care arătam că elementul x aparține mulțimii X îl exprimăm în scris astfel:

$x \in X$ și se citește x aparține mulțimii X; dacă x nu aparține mulțimii X scriem $x \notin X$.

De exemplu, numărul 3 aparținînd mulțimii numerelor naturale îl scriem $3 \in N$; cum 3,7 nu aparține acestei mulțimi, scriem $3{,}7 \notin N$.

Relația $x \in X$ o numim relație de apartenență.

Mulțimea care nu conține nici un element o numim „mulțimea vidă" și o notăm, de obicei, cu $\emptyset$.

Dacă toate elementele mulțimii X aparțin mulțimii Y, dar unele elemente ale mulțimii Y nu aparțin mulțimii X spunem că mulțimea X este o submulțime proprie a mulțimii Y și scriem $X \subset Y$ și citim „mulțimea X este inclusă în mulțimea Y" sau „mulțimea Y include mulțimea X. Relația de mai sus se numește relația de incluziune."

Dacă orice element al mulțimii X este element al mulțimii Y și invers, orice element al mulțimii Y este element și al mulțimii X, spunem că mulțimile coincid și scriem:

$$X = Y.$$

Această proprietate o folosim în general pentru a dovedi egalitatea a două mulțimi, adică din $X \subset Y$ și $Y \subset X$ re-

Scanned with OKEN Scanner

zultă $X = Y$. În scris: dacă $X \subset Y$ și $Y \subset X \Rightarrow X = Y$.

Menționăm de asemenea, incluzinea $\emptyset \cup X$, oricare ar fi mulțimea X.

3°. **Operații cu mulțimi.** Fie E o mulțime și X și Y două submulțimi ale ei. Mulțimea elementelor care aparțin cel puțin uneia din mulțimile X și Y este o nouă mulțime, numită „reuniunea mulțimilor X și Y“ și se notează $X \cup Y$.

Din definiție rezultă

$$X \cup Y = \{ X \text{ sau } x \in X, \text{ sau } x \in Y \}.$$

Semnul $\cup$ îl numim „semnul reuniunii“; $X \cup Y$ îl citim „X reunit cu Y“.

Exemple

1) $X = 2,4,6,\ Y = 1,3,5,6,7,\ X \cup Y = 1,2,3,4,6,7$

2) $X = \{ x \mid x = 2\ K + 1,\ K \in N \} = \{1,3,5,\ldots,\}$

$Y = \{ x \mid x = 2\ K,\ K \in N \} = \{ 2,4,6,\ldots,\}$

$$X \cup Y = N.$$

Menționăm reuniunile:

$$X \cup X = X,\ X \cup \emptyset = X,\ E \cup X = E.$$

Mulțimea elementelor comune mulțimilor X și Y este o nouă mulțime, numită intersecția lui X și Y.

Notăm: $X \cap Y$ și citim „X intersectat cu Y“. Semnul $\cap$ se numește semnul de intersecție. Din definiție rezultă $X \cap Y = \{ X \mid x \in X \text{ și } x \in Y \}$.

Exemplu:

$$X = \{ 1,3,5,6,7,8, \}\ Y = \{ 2,4,5,6,8,9, \},$$

$$X \cap Y = \{ 5,6,8 \}.$$

Menționăm intersecțiile: $X \cap X = X,\ X \cup \emptyset = \emptyset,\ X \cap E = X$.

Scanned with OKEN Scanner

Dacă mulțimile X și Y nu au nici un element comun, adică

$$X \cap Y = \emptyset,$$

spunem că mulțimile X și Y sînt disjuncte.

Exemplu. $X = \{1,3,5,\ldots,\} = \{x \mid x = 2K + 1, K \in N\}$, $Y \{= 2,4,6,\ldots,\} = \{x \mid x = 2K;\ K \in N\}$, $X \cap Y = \emptyset$.

4°. **Produsul cartezian.** Fie X și Y două submulțimi ale unei mulțimi E, distincte sau nu. Mulțimea formată din toate perechile ordonate (x, y), de elemente în care $x \in X$ și $y \in Y$, pe care o notăm $X \times Y$, o numim produsul cartezian al mulțimilor X și Y. Mulțimile X și Y le numim factorii produsului cartezian.

Din definiție rezultă:

$$X \times Y = \{x,\ y \mid x \in X,\ y \in Y\}.$$

Dacă E este o mulțime și X o submulțime proprie a sa, atunci elementele lui E ce nu aparțin lui X formează *complementara* mulțimii X față de mulțimea E și se notează $C_E X$.

Din definiție rezultă:

$$C_E X = \{x \mid x \notin X,\ x \in E\}.$$

Menționăm proprietățile:

$$CC_E X = X,\ C_E E = \emptyset,\ C_E \emptyset = E,$$

$$X \cup C_E X = E, \quad X \cap C_E X = \emptyset.$$

Dacă $X \subset E$, $Y \subset E$, subzistă relațiile $C(X \cup Y) = (C_E X) \cap C_E Y$ și $C_E(X \cap Y) = C_E Y \cup C_E Y$, numite formulele lui De Morgan.

5°. **Diferența a două mulțimi.** Fie X și Y două submulțimi ale unei mulțimi date. Prin diferență a mulțimilor X și Y, pe care o notăm $X - Y$, înțelegem mulțimea elementelor ce aparțin lui X și nu aparțin lui Y, adică:

$$X - Y = \{x \mid x \in X;\ x \notin Y\}.$$

Exemplu: $X = \{2,4,5,7,9,11,\}$, $Y = \{4,5,9,\}$, $X - Y = \{2,7,11\}$.

Scanned with OKEN Scanner

Menționăm relația utilă în aplicații: $X - Y = X \cap C_E Y$.

6°. **Partiția unei mulțimi.** Prin definiție, numim partiția unei mulțimi X o familie de mulțimi $X_1, X_2, \ldots, X_n$, cu proprietățile:

a) Fiecare dintre mulțimile X_i $(i = 1,\ldots,n)$ conținînd cel puțin un element, adică $X_i \neq \emptyset$ $(i = 1,2,\ldots,n)$.

b) Mulțimile X_i $(i = 1,2,\ldots,n)$, sînt două cîte două disjuncte, adică

$$X_i \cap X_j = \emptyset \quad (i \neq j; \quad i,j = 1,2,\ldots,n).$$

c) Reuniunea mulțimilor X_i $(i = 1,\ldots,n)$ este egală cu mulțimea X, adică

$$X = X_1 \cup X_2 \cup \ldots \cup X_n.$$

Vom spune că două elemente x_1 și x_2, aparținînd lui X, sînt „echivalente", dacă ele aparțin unei aceleiași mulțimi X_k din partiția dată. Notăm $x_1 \equiv x_2$.

Dacă d este un întreg oarecare, se spune că numerele întregi a și b sînt echivalente, modulo d, dacă prin împărțirea cu d dau resturi egale, și scriem:

$$a = b \pmod{d}.$$

7°. **Legi de corespondență între mulțimi.** Considerăm două mulțimi X și Y. O lege F, care face să corespundă fiecărui element $x \in X$ un element $y \in Y$, bine determinat, îl numim *aplicație univocă* a lui X în Y. Elementul $y \in Y$ îl numim imaginea lui $x \in X$ prin F.

Dacă legea de corespondență F face ca oricărui element $x \in X$ să-i corespundă o submulțime a lui Y, bine determinată, o numim „aplicație multivocă" a lui X în Y, și se notează F_x.

Observații: În definițiile de mai sus, cele două mulțimi X și Y pot fi distincte.

Fie F o aplicație definită pe X cu valori în X și fie Y o submulțime proprie a lui X, adică $Y \subset X$.

Vom putea scrie relația $F(Y) = \bigcup_{x \in Y} F_x$, unde $F(Y)$ este imaginea submulțimii $x \in Y$, iar semnul $\bigcup$ arată că vom

Scanned with OKEN Scanner

reuni imaginile prin F ale tuturor elementelor $x \in Y$.

Vom defini aplicațiile F^2, F^3,..., F^k astfel:

$$F_x^2 = F(F_x)$$

$$F_x^3 = F(F_x^2),$$

$$\ldots\ldots\ldots\ldots$$

$$F_x^k = F(F_x^{k-1}).$$

Numim *închidere tranzitivă* a lui F aplicația $\hat{F}$ a mulțimii X în ea însăși, definită în modul următor:

$$\hat{F}_x = \{X\} \cup F_x \cup F_x^2 \cup \ldots \cup F_x^k.$$

De fapt, închiderea tranzitivă este o mulțime formată din reuniunea punctului x, mulțimea aplicațiilor de ordin unu a lui x, mulțimea aplicațiilor, aplicații de ordinul unu a lui x, care formează mulțimea de ordinul doi a lui X etc.

Elementele mulțimii aplicațiilor de ordinul k ale punctului x, F_x^k, pe parcursul lucrării vor fi adesea numite „urmașii de ordinul k“ ai punctului x.

Prin aplicație inversă, notată F^{-1}, vom înțelege mulțimea punctelor care au ca aplicație directă punctul x, ceea ce simbolic scriem:

$$F_x^{-1} = \{y \mid x \in F_y\},$$

pe care pe parcurs i-am numit strămoșii lui x.

§ 2. ELEMENTE DE CALCUL MATRICEAL

1°. **Introducere.** Tabloul dreptunghiular format din m linii și n coloane.

$$A = \begin{bmatrix} a_{11} & a_{12} \ldots\ldots a_{1j} \ldots\ldots a_{1n} \\ a_{21} & a_{22} \ldots\ldots a_{2j} \ldots\ldots a_{2n} \\ a_{i1} & a_{i2} \ldots\ldots a_{ij} \ldots\ldots a_{in} \\ a_{m1} & a_{m2} \ldots\ldots a_{mj} \ldots\ldots a_{mn} \end{bmatrix}$$

Scanned with OKEN Scanner

pentru care definim anumite reguli de calcul pe care le precizăm mai departe, poartă numele de matrice de dimensiunile $(m \times n)$.

Elementul așezat la intersecția liniei i cu coloana j îl notăm prin a_{ij}, de pildă elementul situat la intersecția liniei a treia cu coloana șase se notează a_{36}.

Prescurtat, matricea de dimensiune $(m \times n)$ se scrie astfel:

$$A = (a_{ij}),\ i = 1,2,\ldots, m;\ j = 1,2,\ldots, n.$$

Dacă elementele a_{ij} sînt toate nule, matricea se numește matrice nulă.

Dacă $m = n = p$, adică numărul liniilor este egal cu numărul coloanelor și ambele egale cu p, matricea se numește „matrice pătrată" de ordinul p.

În cazul în care $m = 1$, ceea ce înseamnă că matricea conține o singură linie, spunem că avem o matrice-linie. O asemenea matrice se mai numește și „vector-linie". Elementele matricei poartă numele, în acest caz, de „componente ale vectorului". Astfel, un vector-linie cu n componente este de forma $a = (a_1, a_2,\ldots, a_n)$.

Analog, pentru $n = 1$ avem o matrice-coloană sau vector coloană, cu m componente de forma

$$a = \begin{pmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_m \end{pmatrix}.$$

Matricea pătrată particulară în care toate elementele sînt egale cu zero, iar cele situate pe diagonala principală $a_{11}, a_{22}, \ldots, a_{nn}$ egale cu $+1$, poartă numele de *matrice unitate*, de ordinul n și o notăm E_n sau simplu E. Astfel, matricea-unitate de ordinul al treilea este

$$E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Scanned with OKEN Scanner

2°. **Operații cu matrice.** Pe mulțimea matricelor definim următoarele operații: adunarea (numai dacă matricele sînt de aceleași dimensiuni), înmulțirea cu un scalar, înmulțirea a două matrice (cu condiția ca numărul coloanelor din prima matrice să fie egal cu numărul liniilor din a doua matrice).

a) *Adunarea matricelor.* Suma a două matrice $A = (a_{ij})$ și $B = (b_{ij})$ este tot o matrice $C = (c_{ij})$ ale cărei elemente se obțin după regula:

$$c_{ij} = a_{ij} + b_{ij},$$

adică elementele matricei-sumă se obțin adunînd elementele corespunzătoare din matricele A și B. Rezultă deci că A și B, trebuie să aibă aceeași dimensiune.

b) Produsul dintre scalarul λ, cu matricea $A = (a_{ij})$ $(i = 1,2, \ldots, m;\ j = 1,2, \ldots, n)$ este o matrice $P = (p_{ij})$ $(i = 1,2, \ldots, m;\ j = 1,2, \ldots, n)$, ale cărei elemente se obțin înmulțind fiecare element al matricei A cu scalarul λ, adică

$$p_{ij} = \lambda a_{ij}\ ,\ (i = 1,2, \ldots, m;\ j = 1,2, \ldots, n).$$

c) *Produsul matricelor.* Fiind date matricele $A = (a_{ik})$ $(i = 1,2, \ldots, m;\ j = 1,2, \ldots, n)$ și $B = (b_{ej})$ $(e = 1,2, \ldots, n;\ j = 1,2, \ldots, p)$.

Produsul lor este prin definiție matricea

$$C = A \times B = (c_{ij})\ \ (i = 1,2, \ldots, m;\ j = 1,2, \ldots, n),$$

al cărei element c_{ij} , situat la intersecția liniei i cu coloana j este egal cu suma produselor elementelor liniei i din matricea A, prin elementele coloanei j din matricea B, adică:

$$c_{ij} = \sum_{k=1}^{n} a_{ik}\, b_{kj}\ \ (i = 1,2, \ldots, n;\ j = 1,2, \ldots, p).$$

Așadar, produsul matricelor A și B are sens numai dacă A este de dimensiune $(m \times n)$, iar B, de dimensiunea $(n \times p)$. Matricea-produs are dimensiunea $(n \times p)$.

Observații: Produsul de matrice în general nu este comutativ

$$A \cdot B \neq B \cdot A,$$

Scanned with OKEN Scanner

3°. **Determinanți.** Fiecărei matrice pătrate, de ordin i, îi asociem un număr numit determinantul matricei respective. Astfel, matricei pătrate de ordinul n îi asociem numărul

$$D_n = \begin{Vmatrix} a_{11}\ a_{12}\ldots a_{1j}\ \ldots\ldots a_{1n} \\ a_{21}\ a_{22}\ldots a_{2j}\ \ldots\ldots a_{2n} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_{i1}\ a_{i2}\ldots a_{ij}\ \ldots\ldots a_{in} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1}\ a_{n2}\ldots a_{np}\ldots\ldots a_{nn} \end{Vmatrix}$$

În particular, pentru $n = 1$ avem determinantul de ordinul întîi $D_1 = a_{11}$, format dintr-un singur element; pentru $n = 2$ avem

$$D_2 = \begin{Vmatrix} a_{11}\ a_{12} \\ a_{21}\ a_{22} \end{Vmatrix}$$

format din $2 \times 2 = 2^2$ elemente. Determinantul de ordinul n, D_n, va avea evident n^2 elemente aranjate pe n linii și n coloane. Dacă în D_n suprimăm linia i și coloana j, obținem un determinant de ordinul $(n - 1)$, care se numește minorul elementului a_{ij} din D_n, pe care-l notăm prin D_{ij}; evident D_n avînd n^2 elemente, va avea în n^2 minori.

Dacă introducem notația

$$A_{ij} = (-1)^{i+j} D_{ij},$$

atunci A_{ij} se numește complementul algebric al elementului a_{ij} din D_n.

Numărul D_n, adică valoarea determinantului de ordinul n îl obținem după regula următoare:

1) $D_1 = |a_{11}| = a_{11}$,

2) $$D_2 = \begin{Vmatrix} a_{11}\ a_{12} \ldots a_{1j} \ldots a_{1n} \\ a_{21}\ a_{22} \ldots a_{2j} \ldots a_{2n} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_{i1}\ a_{i2} \ldots a_{ij} \ldots a_{in} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1}\ a_{n2} \ldots a_{nj}\ ,\ldots\ a \end{Vmatrix} = a_{11}A_{11} + a_{12}A_{12} + \ldots + a_{1j}A_{1j} + \ldots + a_{1n}A_{1n}$$

Așadar, valoarea determinantului de ordinul întîi (format dintr-un singur element) este egală cu elementul însuși. Valoarea unui determinant de ordinul n este egală cu suma

Scanned with OKEN Scanner

produselor dintre elementele primei linii și complementii algebrici corespunzători. Se spune că am dezvoltat determinantul după elementele primei linii prin regula minorilor. Se poate însă demonstra că dacă dezvoltăm determinantul după oricare linie sau oricare coloană, obținem același rezultat, deci putem scrie,

$$D_n = \begin{cases} \sum_{j=1}^{n} a_{ij} A_{ij} \text{ dezvoltare după linia } i, \\ \sum_{j=1}^{n} a_{ij} A_{ij} \text{ dezvoltare după coloana } j. \end{cases}$$

De aici decurg o serie de proprietăți care ușurează calculul valorii unui determinant.

4°. **Inversa unei matrice.** Dacă A este o matrice pătrată de ordinul n, al cărei determinant este diferit de zero, numim inversa ei o nouă matrice A^{-1}, care se bucură de proprietatea că înmulțită la dreapta sau la stînga cu matricea dată, conduce la matricea-unitate de ordinul n, adică

$$AA^{-1} = A^{-1}A = E.$$

Pentru determinarea matricei inverse, parcurgem următoarele etape:

a) Calculăm determinantul matricei, pe care-l notăm prin $|A|$; dacă acesta este diferit de zero, are sens să continuăm calculele, dacă nu, matricea dată nu admite inversă.

b) Calculăm „transpusa" matricei A, care o obținem din matricea dată prin schimbarea liniilor în coloane. Deci dacă

$$A = (a_{ij})(i, j = 1,2, \ldots, n),$$

atunci transpusa $\bar{A} = (a_{ji})(i, j = 1,2, \ldots, n)$.

c) Calculăm „asociata" matricei $\bar{A}$ ale cărei elemente sînt complementi algebrici ai elementelor a_{ji}, notată cu A^*.

d) Inversa lui A o obținem împărțind elementele lui A^* prin determinantul $|A|$.

Vom avea deci,

$$A^{-1} = \frac{1}{|A|} \cdot \bar{A}^*.$$

Scanned with OKEN Scanner

Verificăm ușor relația de definiție $A^{-1} \cdot A = AA^{-1} = E$.

Unele matrice speciale (booleene, latine) au fost definite și prezentate în măsura în care ne-au fost necesare în cuprinsul lucrării.

§ 3. NOȚIUNI DE TEORIA PROBABILITĂȚILOR

1°. **Evenimente aleatoare.** Orice experiență practică conduce la diverse rezultate, numite evenimente. În cele ce urmează ne vom ocupa de o categorie specială de evenimente, și anume de acele evenimente a căror realizare depinde de întîmplare și cărora le vom spune evenimente aleatoare.

Exemple.

a) Dacă experiența constă din aruncarea unei monede, putem realiza fiecare din evenimentele:

E_1 = apariția stemei,
E_2 = apariția banului.

b) Aruncînd un zar, se poate realiza oricare din evenimentele:

E_1 = apariția feței cu un punct,
E_2 = apariția feței cu două puncte,
.
.
.
E_6 = apariția feței cu șase puncte,

c) Dacă într-un lot de 30 de piese 28 sînt corespunzătoare și două defecte, o experiență care constă din extragerea la întîmplare a unei piese din lot poate da naștere următoarelor evenimente:

$E_1, E_2, .., E_{28}$ = extracția unei piese bune,
E_{29}, E_{30} = extracția unei piese defecte.

d) Dacă experiența constă din alegerea la întîmplare a unui punct $x \in [0,1]$, putem obține ca rezultat al acestei experiențe o infinitate de evenimente.

Scanned with OKEN Scanner

Evenimentele obținute în exemplele date mai sus, ca rezultat direct al experienței, le vom spune *evenimente elementare*. Alături de aceste evenimente vom considera și alte evenimente. De pildă, în experiența din primul exemplu putem considera evenimentul care constă din apariția oricăreia din cele două fețe și pe care-l vom nota prin E. Este evident că în oricare experiență acest eveniment se produce în mod sigur, de aceea îl vom numi *evenimentul sigur*. Dacă, din contra, considerăm evenimentul ca dintr-o aruncare cu banul să nu apară nici una din fețe, obținem un rezultat imposibil, de aceea acestui eveniment i se spune *evenimentul imposibil*, pe care-l vom nota prin $\emptyset$.

În general, date fiind două evenimente oarecare, într-o experiență ne interesează de pildă realizarea unuia sau altuia, indiferent care, sau realizarea lor simultană. În primul caz, vom scrie $A \cup B$ și înțelegem că acest eveniment, numit *reuniunea* evenimentelor A și B se realizează dacă se realizează sau A, sau B; în al doilea caz, vom scrie $A \cap B$, și acest eveniment, numit intersecția evenimentelor A și B se realizează dacă ambele se realizează simultan. Așa, de pildă, în exemplul al treilea, dacă experiența constă din extracția la întîmplare a două piese din lot și notăm prin α evenimentul care constă în extracția a două piese de calități diferite, realizarea lui constă din realizarea oricăruia din evenimentele $E_i \cup E_j$, unde $i = 1, 2, \ldots, 28$, iar $j = 29, 30$.

Așadar, considerînd o anumită experiență care se efectuează în condiții bine determinate, ea poate să conducă la: a) o mulțime de evenimente elementare pe care o notăm $\{E_i\}$ $(i = 1, 2, \ldots, n)$, b) la evenimentul sigur, care constă din apariția oricăruia din cele n evenimente elementare, deci

$$E = E_1 \cup E_2 \cup \ldots \cup E_n,$$

c) la evenimentul imposibil $\emptyset$ și în sfîrșit, d) la evenimente complexe de forma

$$A = E_1 \cup E_2 \cup \ldots \cup E_k \quad (k \leqslant n).$$

Aceste evenimente sînt în număr de C_n^k, evenimentul A realizîndu-se atunci cînd se realizează sau E_1, sau E_2, ..., sau

Scanned with OKEN Scanner

E_k, adică indiferent care din ele. În acest ultim caz spunem că experiența efectuată generează un cîmp de evenimente pe care-l notăm prin $[E, K]$. Numărul evenimentelor cîmpului este egal cu

$$1 + \sum_{k=1}^{n} C_n^k = 2^n.$$

Definiție. Evenimentul A^*, care constă în nerealizarea evenimentului A, se numește opusul lui A. Deoarece aceste evenimente nu se pot realiza simultan, vom scrie, pe de o parte,

$$A \cap A^* = \emptyset,$$

iar pe de altă parte, realizarea unuia din ele fiind sigură (deoarece, dacă nu se realizează A, se realizează A^* și reciproc), vom avea

$$A \cup A^* = E.$$

În general, două evenimente A și B care nu se pot realiza simultan se numesc evenimente incompatibile între ele și sînt caracterizate prin relația

$$A \cap B = \emptyset,.$$

Spunem că mulțimea de evenimente $A_1, A_2, \ldots, A_k$ este o *desfacere* a evenimentului sigur, dacă sînt incompatibile două cîte două și reuniunea lor este egală cu evenimentul sigur, adică

$$A_1 \cup A_2 \cup \ldots \cup A_k = E,\ A_i \cap A_j = \emptyset\ (i, j = 1, 2, \ldots, k; i \neq j).$$

Mulțimea $\{E_i\}$ $(i = 1, 2, \ldots, n)$ a evenimentelor elementare constituie o desfacere a evenimentului sigur, deoarece:

$$E_1 \cup E_2 \cup \ldots \cup E_n = E,\ E_i \cap E_j = \emptyset\ (i, j, = 1, 2, \ldots, n; i \neq j).$$

Din cele arătate pînă aici, cîmpul $[E, K]$ are structura următoare:

$$[E, K] = \{\emptyset, \{E_i\}, \{E_i \cup E_j\}, \{E_i \cup E_j \cup E_k\}, \ldots, \{E_1 \cup E_2 \cup \ldots \cup E_n\} = E\}.$$

Scanned with OKEN Scanner

De exemplu, cîmpul generat de experiența constînd în aruncarea unei monede are $2^n = 2^2 = 4$ evenimente

$$[E, K] = \{\emptyset, E_1, E_2, E_1 \cup E_2 = E\}.$$

2°. **Probabilități.** Vom înțelege prin probabilitatea unui eveniment $X \in [E, K]$ o funcție P, definită pe cîmpul de evenimente $[E, K]$ verificînd următoarele axiome:

a) $P(X) \geqslant 0$, $X \in [E, K]$,

b) $P(E) = 1$; posibilitatea evenimentului sigur este egală cu unitatea

c) $P(X \cup Y) = P(X) + P(X)$,
$X \in [E, K]$ și $Y \in [E, K]$ cu $X \cap Y = \emptyset$,

adică probabilitatea reuniunii a două evenimente incompatibile este egală cu suma probabilităților lor.

Observații. Din axioma c) deducem $P(\emptyset) = 0$, adică probabilitatea evenimentului imposibil este egală cu zero.

Evenimentului $A \in [E, K]$ îi corespunde numărul $P(A)$, probabilitatea sa. Prin definiție, mulțimea evenimentelor cîmpului în corespondența cu probabilitățile lor formează un cîmp de probabilități $[E, K, P]$.

Axioma c) o putem generaliza la n evenimente incompatibile două cîte două:

$$P\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{i=1}^{n} P(A_i),$$

unde

$$\bigcup_{i=1}^{n} A_i = A_1 \cup A_2 \cup \ldots \cup A_n, \quad A_i \cap A_j = \emptyset \quad (i \neq j).$$

În cazul în care evenimentele sînt compatibile, probabilitatea reuniunii a n evenimente este dată de formula:

$$P\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{i=1}^{n} P(A_i) - \sum_{i \neq j}^{n} P(A_i \cap A_j) + \sum_{\substack{i \neq j \neq k \\ i,j,k=1}}^{n} P(A_i \cap A_j \cap A_k) +$$

$$+ \ldots + (-1)^{n-1} P\left(\bigcap_{i=1}^{n} A_i\right).$$

Scanned with OKEN Scanner

Observație: Dacă evenimentele A_i sînt incompatibile două cîte două, $P(A_i\, A_j \ldots) = P(\emptyset) = 0$, obținem formula de mai sus de adunarea probabilităților.

Probabilități condiționate. Fie evenimentele X și Y definite pe același cîmp $[E, K]$. Numim probabilitatea lui X condiționată de Y și o notăm $P(X,Y)$ sau $P_y(X)$, expresia

$$P_y(X) = \frac{P(X \cap Y)}{P(Y)}.$$

Putem arăta că această probabilitate verifică axiomele a), b), c). Din formula de mai sus putem scrie:

$$P(X \cap Y) = P(Y) \cdot P_y(X).$$

Definiție. Dacă $P_y(X) = P(X)$, adică dacă probabilitatea realizării evenimentului X nu depinde de realizarea evenimentului Y, spunem că evenimentul X este *independent de evenimentul* Y.

Observație: Deoarece putem scrie
$P(X \cap Y) = P(X)P_x(Y) = P(Y)P_y(X)$ și dacă $P_y(X) = P(X)$, rezultă $P_x(Y) = P(Y)$, deci și Y este independent de X.

Evenimentele X și Y, pentru care există relația

$$P(X \cap Y) = P(X) \cdot P(Y),$$

se numesc evenimente independente și relația de mai sus dă probabilitatea realizării lor simultane.

Înmulțirea probabilităților. Probabilitatea producerii simultane a evenimentelor independente A_i $(i = 1, \ldots, n)$ este egală cu produsul probabilității acestora, adică

$$P\left(\bigcap_{i=1}^{n} A_i\right) = \prod_{i=1} P(A_i)$$

unde

$$\bigcap_{i=1}^{n} A_i = A_i \cap A_2 \cap \ldots \cap A_n.$$

3°. **Variabile aleatoare.** După cum știm, o lege care asociază fiecărui element A un element unic $y \in B$ definește

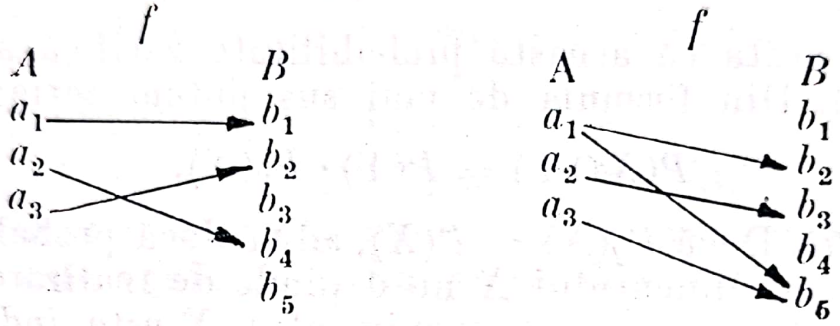
Scanned with OKEN Scanner

o funcție pe mulțimea A cu valori în mulțimea B. Notînd prin f legea de corespondență de la mulțimea A la mulțimea B, de obicei scriem:

$$f : A \to B$$

și citim „funcția f definită pe A cu valori în B".

Dacă, de exemplu, $A = (x_1, x_2, x_3)$, $B = (b_1, b_2, b_3, b_4, b_5)$ și definim corespondențele:



atunci prima, conform definiției, reprezintă o funcție definită pe A cu valori în B, în timp ce a doua nu reprezintă o funcție, deoarece elementului $a_1 \in A$ nu-i corespunde un element unic în B, ci două: b_2 și b_5.

Am văzut mai înainte că o experiență generează un cîmp de evenimente $[E, K]$, conținînd 2^n elemente, unde n este numărul evenimentelor sale elementare $E_1, E_2, \ldots, E_n$. Dacă fiecărui eveniment elementar îi vom pune în corespondență un anumit număr dintr-o mulțime de numere, după o regulă oarecare, definim evident o funcție, să-i spunem $X(e)$ pe mulțimea $\{E_i\}$ a evenimentelor elementare, cu valori în mulțimea arbitrară de numere reale, pe care s-o notăm prin B.

Exemple:

1) Experiența constînd din aruncarea unei monede, ne conduce la funcția,

$$\begin{array}{l} \overset{X(e)}{E_i \longrightarrow B} \\ E_1 \longrightarrow 0 \\ E_2 \longrightarrow 1 \end{array} \quad \text{adică } X = X(e) = \begin{cases} 0, \text{ dacă } e = E_1, \\ 1, \text{ dacă } e = E_2. \end{cases}$$

Avem, evident,

$$P(E_1) = P(X = 0) = \frac{1}{2},\ P(E_2) = P(X = 1) = \frac{1}{2},$$

de unde

$$P(E_1) + (P(E_2) = 1.$$

2) Experienţa constînd din aruncarea unui zar, ne poate conduce la funcţia

$$X = X(e) = i \text{ pentru } e = \dot{E}_i \quad (i = 1, 2, 3, 4, 5, 6),$$

şi aici

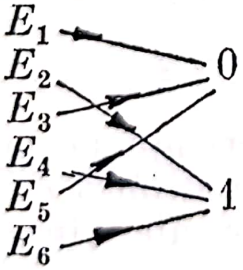
$$\sum_{i=1}^{6} P(E_i) = \sum_{i=1}^{6} P(X = i) = 1.$$

Dacă în aceeaşi experienţă notăm prin A_1 apariţia feţei cu număr fără soţ de puncte şi prin A_2 apariţia feţei cu număr par de puncte, avem evident:

$$A_1 = E_1 \cup E_3 \cup E_5, \qquad A_2 = E_2 \cup E_4 \cup E_6,$$
$$A_1 \cup A_2 = E \qquad A_1 \cap A_2 = \emptyset.$$

Asociind evenimentului A_1 numărul 0 şi lui A_2 pe 1, atunci $B = \{0,1\}$, şi obţinem funcţia

$$\{E_i\} \xrightarrow{X(e)} B$$



$$X = X(e) = \begin{cases} 0, \text{ dacă } e = E_1, E_3, E_5 \\ 1, \text{ dacă } e = E_2, E_4, E_6. \end{cases}$$

Vom putea scrie, de asemenea,

$$P(A_1) = P(X = 0)$$

sau încă

$$P(A_1) = P(E_1 \cup E_2 \cup E_5) = P(E_1) + P(E_3) + P(E_5) = \frac{3}{6}.$$

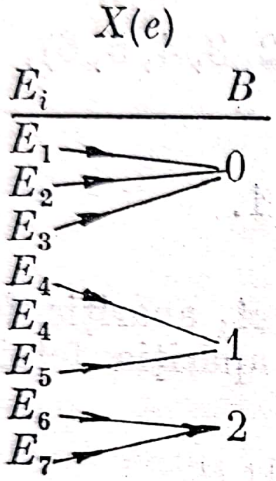
Scanned with OKEN Scanner

În mod analog, găsim:

$$P(A_2) = P(X = 1) = \frac{3}{6}$$

de unde $P(A_1) + P(A_2) = 1$

3) Presupunem că într-o urnă sînt 7 bile, din care 3 albe, 2 albastre și 2 roșii. Notînd cu A_1, A_2, A_3 evenimentele de apariție într-o extracție a unei bile din urnă, a unei bile albe, albastre sau roșii, putem avea corespondența:



$$X = X(e) = \begin{cases} 0, \text{ dacă } e = E_1, E_2, E_3, \\ 1, \text{ dacă } e = E_4, E_5, \\ 2, \text{ dacă } e = E_6, E_7. \end{cases}$$

Asociind evenimentelor A_1, A_2, A_3, respectiv valorile 0, 1, 2 putem scrie:

$$P(A_1) = P(X = 0) = \frac{3}{7}$$

$$P(A_2) = P(X = 1) = \frac{1}{7}$$

$$P(A_3) = P(X = 2) = \frac{2}{7}$$

$$\sum_{i=1}^{3} P(A_i) = \frac{3}{7} + \frac{2}{7} + \frac{2}{7} = 1.$$

Din aceste exemple deducem că o experiență ne poate conduce la o funcție $X = X(e)$, definită pe mulțimea $E_i (i = 1, 2, \ldots, n)$ a evenimentelor elementare ale cîmpului $[E, K]$ generate de această experiență.

Vom nota prin $x_1, x_2, \ldots, x_k$ $(k \leqslant n)$ sînt valorile funcției corespunzînd evenimentelor $A_1, A_2, \ldots, A_k$, unde $\sum_{i=1}^{k} A_i = E$ și $P(A_i) = p_i$ deci $\sum_{i=1}^{k} p_i = 1$.

Definiție. Funcția $X = X(e)$ o numim *variabilă aleatoare* (statistică sau stochastică), definită pe cîmpul $[E, K, P]$.

Vom scrie astfel:

$$X \begin{pmatrix} x_1, x_2, \ldots, x_k \\ p_1, p_2, \ldots, p_k \end{pmatrix}$$

sau mai simplu:

$$X \begin{pmatrix} x_i \\ p_i \end{pmatrix}_{i = 1, 2, \ldots, k}.$$

Definiție. Mulțimea valorilor în corespondență cu probabilitățile respective ale lui X se numește *distribuție* sau *repartiția* variabilă aleatoare X.

Operații cu variabilele aleatoare

Suma variabilelor aleatoare

$$X \begin{pmatrix} x_i \\ p_i \end{pmatrix}_{i = 1, \ldots, m}, \qquad Y \begin{pmatrix} y_j \\ p_j \end{pmatrix}_{j = 1, \ldots, n}$$

este variabila

$$X + Y \begin{pmatrix} x_i + y_j \\ p_{ij} \end{pmatrix}_{i = 1, \ldots, m \; ; \; j = 1, \ldots, n},$$

unde p_{ij} este probabilitatea ca simultan variabila X să ia valoarea x_i, iar variabila Y să ia valoarea Y_j, adică $p_{ij} = P\,(X = x_i \cap Y = y_j)$.

Produsul variabilelor X și Y este variabila aleatoare

$$XY \begin{pmatrix} x_i y_j \\ p_{ij} \end{pmatrix}_{i = 1, \ldots, m; \; j = 1, \ldots, n}.$$

În particular, vom avea distribuțiile:

$$aX \begin{pmatrix} aX_i \\ p_i \end{pmatrix}_{i=1, \ldots, m}, a + X \begin{pmatrix} a + X_i \\ p_i \end{pmatrix}_{i=1, \ldots, m}, X^r \begin{pmatrix} x_i^r \\ p_i \end{pmatrix}_{i=1, \ldots, m}.$$

Momentele variabilităților aleatoare. Prin definiție, expresia

$$m_r = M(X^r) = \sum_{i=1}^{n} p_i x_i^r, \quad \sum_{i=1}^{n} p_i = 1,$$

Scanned with OKEN Scanner

se numește momentul de ordinul r al variabilei

$$X\begin{pmatrix} x_i \\ p_i \end{pmatrix}_{i=1,\ \ldots,\ n},$$

iar

$$m_r^+ = M(|X|^r) = \sum_{i=1}^{n} p_i |x_i|^r,$$

se numește momentul absolut de ordinul r al variabilei aleatoare X.

În particular pentru $r = 1$ obținem valoarea medie sau speranța matematică:

$$m = M(X) = p_1 \cdot x_1 + p_2 \cdot x_2 + \ldots + p_n \cdot x_n,$$

cu proprietățile cunoscute:

1) $M(X + Y) = M(X) + M(Y)$
2) $M(X, Y) = M(X)\, M(Y)$,

dacă variabilele X și Y sînt independente.

Expresia,

$$M_r = \sqrt[r]{m_r} = (\Sigma\, p_i x_r^i)^{\frac{1}{r}}$$

o numim valoarea medie de ordinul r a variabilei X.

O proprietate importantă a acestor valori medii este aceea că pentru $x > 0$ avem $M_r \leqslant M_s$, dacă $r \leqslant s$.

Cazuri particulare:

1) Pentru $r = 1$ obținem media

$$m = M(X) = \sum_{i=1}^{r} p_i x_i.$$

Notînd $p_i = \dfrac{\lambda_i}{\sum\limits_{i=1}^{n} \lambda_i^m}$, de unde $\sum\limits_{i=1}^{n} p_i = \sum\limits_{i=1}^{n} \dfrac{\lambda_i}{\sum\limits_{i=1}^{n} i} = 1$, găsim

media aritmetică ponderată

$$M = \frac{\sum\limits_{i=1}^{n} \lambda_i\, x_i}{\sum\limits_{i=1}^{n} \lambda_i}$$

Scanned with OKEN Scanner

a valorilor x_i cu ponderile λ_i $(i = 1, \ldots, m)$.

2) Pentru $r = -1$ avem media

$$M_{-1} = \frac{1}{\sum_{i=1}^{n} \frac{p_i}{x_i}}$$

din care deducem media armonică cunoscută

$$H = \frac{\sum_{i=1}^{n} \lambda_i}{\sum_{i=1}^{n} \frac{\lambda_i}{x_i}}.$$

3) Pentru $r = 0$ găsim media $M_0 = x_1^{p_1} x_2^{p_2} \ldots x_m^{p_m}$, de unde media geometrică obișnuită,

$$G = \sqrt[\Sigma\lambda_i]{x_1^{\lambda_1}, \ldots, x_n^{\lambda_n}}.$$

În baza proprietății menționate, obținem relația de ordine dintre cele trei medii:

$$H \leqslant G \leqslant M,$$

relație care se poate de altfel dovedi în mod direct.

Momente centrate. Prin definiție, momentul de ordinul r al variabilei aleatoare

$$X - M(X) = X - m$$

îl numim momentul centrat de ordinul r al variabilei X. În baza acestei definiții, putem deci scrie, notînd acest moment prin

$$M_r = M\,[(X - m)^r] = \sum_{i=1}^{n} p_i (x_i - m)^r.$$

Un indicator numeric care măsoară împrăștierea valorilor lui X față de valoare sa medie $M(X) = m$ este *dispersia* (sau varianța) variabilei X și care este momentul său central de ordinul al doilea

$$M_2 = D^2(X) = \sum_{i=1}^{n} p_i (x_i - m)^2.$$

Scanned with OKEN Scanner

Notăm de obicei $M_2 = \sigma^2$. Valoarea $\sqrt{M_2} = D(X) = \sigma$ o numim *abaterea medie pătratică* a variabilei X.

4°. **Lanțuri Markov. Matrice stochastice.** Deoarece teoria matricelor stochastice joacă un rol de bază în teoria proceselor markoviene, vom expune la început unele noțiuni și rezultate cu privire la această categorie de matrice speciale.

O matrice pătrată

$$A = \begin{pmatrix} a_{11}\ a_{12}, \ldots, a_{1n} \\ a_{21}\ a_{22}, \ldots, a_{2n} \\ \ldots\ldots\ldots\ldots \\ a_{n1}\ a_{n2}, \ldots, a_{nn} \end{pmatrix} = (a_{ij})_{ij = 1,2,\ldots,n}$$

o numim *nenegativă* dacă pentru orice i și j avem $a_{ij} \geqslant 0$; vom scrie $A \geqslant 0$. Matricea A este pozitivă și o scriem $A > 0$, dacă pentru orice i și j, $a_{ij} > 0$.

Definiție. Matricea nenegativă

$$P = \begin{pmatrix} p_{11}\ p_{12}, \ldots, p_{1n} \\ p_{21}\ p_{22}, \ldots, p_{2n} \\ p_{n1}\ p_{n2}, \ldots, p_{nn} \end{pmatrix}$$

o numim *matrice stochastică*, dacă îndeplinește condițiile:

1) $\sum_{j=1}^{n} p_{ij} = 1,\ i = 1, 2, \ldots, n.$

2) Nici o linie sau nici o coloană a matricei A nu este vidă (adică nu are toate elementele egale cu zero).

Și pentru matricele stochastice putem avea $P > 0$ sau $P < 0$, iar definiția este aceeași ca mai sus.

Introducînd matricea unitate de ordinul n,

$$E = \begin{pmatrix} 1\ 0, \ldots, 0 \\ 0\ 1, \ldots, 0 \\ \ldots\ldots\ldots \\ \ldots\ldots\ldots \\ 0\ 0, \ldots, 1 \end{pmatrix},$$

Scanned with OKEN Scanner

ecuația

$$P(\lambda) = |\lambda E - P| = 0$$

sau dezvoltat

$$P(\lambda) = \begin{vmatrix} \lambda - p_{11} & -p_{12} & \cdots & -p_{1n} \\ -p_{21} & \lambda - p_{22} & \cdots & -p_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ -p_{n1} & -p_{n2} & \cdots & \lambda - p_{nn} \end{vmatrix} = 0,$$

o numim ecuația caracteristică a matricei stochastice P, iar rădăcinile ei le numim numere caracteristice ale matricei.

O teoremă importantă, datorită lui Frebenius, afirmă că ecuația caracteristică $P(\lambda) = 0$ are rădăcina simplă $\lambda_0 = 1$ și pentru oricare altă rădăcină λ_i $(i \neq 0)$, $|\lambda_i| \leqslant 1$, dacă $P > 0$.

Condiția necesară și suficientă ca rădăcina $\lambda_0 = 1$ să aibă ordinul k de multiplicitate este ca toți minorii principali de ordinul $n - 1$, $n - 2$, ..., $n - k + 1$ să fie nuli.

Studiul sistematic al lanțurilor Markov simple și discrete a fost făcut pe baza proprietăților rădăcinilor ecuației matricei stochastice P din care reproducem numai cîteva.

Definiție. Matricea P este decompozabilă, dacă printr-o permutare identică a liniilor și coloanelor poate fi scrisă sub forma

$$P = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix},$$

unde α, β, γ, δ sînt submatrice ale lui P; α, δ sînt matrice pătrate, iar β și γ matrice dreptunghiulare, din care una cel puțin este nulă (adică toate elementele ei sînt egale cu zero).

În legătură cu aceste matrice enunțăm:

Teorema I. Dacă unul din minorii principali ai determinantului $P(\lambda)$ este nul, atunci matricea $P(\lambda)$ este decompozabilă.

Scanned with OKEN Scanner

Teorema II. Condiția necesară și suficientă ca rădăcina $\lambda_0 = 1$ să fie de ordinul de multiplicitate k este ca matricea P să poată fi pusă sub forma

$$P = \begin{pmatrix} P_{11} & 0 & 0, \ldots, 0 \\ P_{21} & P_{22} & 0, \ldots, 0 \\ \ldots\ldots\ldots\ldots\ldots \\ P_{m1} & P_{m2} & \ldots P_{mm} \end{pmatrix}$$

unde $k \leqslant m$, iar P_{ij} sînt submatrice ale matricei P.

Problema variabilelor aleatoare înlănțuite constituie în momentul de față unul din capitolele cele mai însemnate ale teoriei probabilităților. Pe lîngă interesul teoretic pe care-l prezintă, această teorie are mari aplicații practice. În cele ce urmează ne ocupăm de procesele Markov.

Fie $\{X_1, X_2, \ldots, X_m\}$ un șir de variabile aleatoare, fiecare din ele putînd lua un număr finit de valori din mulțimea $E = \{a_1, \ldots, a_m\}$, numită mulțimea stărilor; dacă este satisfăcută relația

$$P(X_n = a_{jn} \mid X_{n-1} = a_{jn-1} \ldots X_1 = a_{j_1}) = P(X_n = a_{jn} \mid X_{n-1} = a_{jn-1})$$

spunem că variabilele $(X_n)_{n \in N}$ formează un lanț Markov simplu.

Deoarece mulțimea E este finită, spunem că lanțul este finit.

Probabilitatea $P(X_1 = a_j) = p_j$, $1 \leqslant j \leqslant m$, avînd proprietățile

$$0 \leqslant p_j \leqslant 1, \; j = 1, \ldots, m,$$

$$\sum_{j=1}^{m} p_j = 1$$

formează repartiția inițială a lanțului Markov considerat mai sus.

Probabilitățile $P(X_n = a_j / X_{n-1} = a_i) = p_{ij}$, unde evident

$$p_{ij} \geqslant 0, \; \sum_{j=1}^{m} p_{ij} = 1, \; i = 1, \ldots, m,$$

Scanned with OKEN Scanner

reprezintă probabilităţile de trecere din starea a_i la momentul $(n-1)$, în stare a_j la momentul n.

Aceste probabilităţi sînt elementele unei matrice stochastice $T = p_{ij}$ $(i, j = 1, \ldots, m)$ numită matricea probabilităţilor de trecere.

Dacă probabilităţile p nu depind de momentele n sau $(n-1)$, adică în general nu depind de timp, procesul îl numim staţionar.

În cazul în care puterile succesive ale matricei t coincid, procesul este constant, adică

$$T^n = T \text{ oricare ar fi } n \in N.$$

O problemă de bază cu privire la lanţurile Markov este aceea a ergodicităţii lor.

Definiţie. Spune că un lanţ Markov este „ergodic", dacă există limita

$$\lim_{n\to\infty} T^n = Q.$$

Matricea Q corespunde stării stabile a lanţului. Liniile acestei matrice sînt identice, ceea ce arată că probabilităţile de trecere nu depind de starea iniţială.

Pe de altă parte, avem:

$$\lim_{n\to\infty} T^n = \lim_{n\to\infty} T^{n+1} = \lim_{n\to\infty} T^n \cdot T = Q,$$

deci rezultă relaţia $QT = Q$.

Această relaţie, împreună cu condiţia ca matricea Q să fie o matrice stochastică, ne dă procedeul analitic cu care putem determina elementele acestei matrice.

Limita de mai sus nu există însă întotdeauna. În acest caz, calculăm limita.

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} T^i = Q_1.$$

Dacă există limita Q_1, spunem de asemenea, că lanţul Markov este ergodic.

Scanned with OKEN Scanner

BIBLIOGRAFIE

1. Avondo — Bodino G., *Economic applications of the Theory of Graphs*, ed. Gordon and Breach, New York, 1962.
2. Altăr I., Miasnico V., M. Rădulescu, D. V. Rădulescu, *Aplicații ale metodelor matematice în probleme de repartiție*, Editura Științifică, 1967.
3. Aurian I., Gh. Boldur, S. Lazăr, *Cercetarea operațională în construcții*, Editura științifică, București, 1967.
4. Berge C., A. Ghanila-Houri, *Programmes, jeux et reseaux de transport*, Dunod, Paris, 1962.
5. Berge C., *Théorie des graphes et ses applications*, Dunod, Paris, 1963.
6. Idem, *Sur l'equivalence du problème du transport généralisé et du problème des réseaux*, C. R. Ac. Science, 1960.
7. Bellman R., *The Theory of dynamic programming*, Bull. A.M.S. 1954.
8. Camion P., *Quelques propriétés des chemins et des circuits, hamiltoniens dans la théorie des graphes*, în „Cahiers du centre d'Etudes de Rech", Belgia (2), 1960.
9. Chow Y., *Théorie et application des graphes de transfert*. Dunod, Paris, 1965.
10. Ciucu Gh., *Elemente de teoria probabilităților și statistica matematică*, Editura didactică și pedagogică, București, 1963.
11. Clark, Charles E., *The P.E.R.T.S. model for distribution of au activity time*, Op. Res., volume 10, nr. 3, 1962.
12. Dantzig G. B., *On the shortest Route through a Network*, Manag. Sc., *6*, 1966.
13. Dantzig G. B., *Finding a cycle ind a graph with minimum cost to timpe ratio with applications a ship Rauting problem*, Op. Rech, 1967.

Scanned with OKEN Scanner

14. Desbazeille G., *Exercices et problèmes de recherche opérationnelle*, Dunod, Paris, 1964.
15. Erdos P., Renyi A., *On the evolutions of random graphs*, Publications Math. Hug. Acad. Sci., 1960.
16. Ford L. R., *A network flow feasibility theorem and combinatorial applications*, T. Math., Canad., 1959.
17. Flood M. M., *On the Hitchcock distribution problem*, Pac. T. Math. 1953/3 a.
18. Pirault M., *Initiation de la quene stationnaire et nonstationnaire*, U.R.S.S., 1939.
19. Gleyzal A., *An algorithm for solving the transportations problem*, Res. Nat. Bur. of Standard, 1965.
20. Ghuila — Hauri A., *Recherche du flot maximum dans certains réseau lorsqu'on impose une condition de bouclage*, London, 1960.
21. Gordon P., *Théorie des chaînes de Markov fines et ses applications*, Dunod, Paris, 1965.
22. Howard R. A., *Dynamic programming and Markov processes*, T. Wilei, New York, 1960.
23. Kaufmann A., *Méthodes et modèles de la recherche opérationnelle*, vol. I și II, Dunod, Paris, 1964.
24. Kaufmann A., Cruon R., *Les fénomènes d' attente*, Dunod, 1961.
25. Kaufmann A., Faure R., *Invitation à la recherche opérationnelle*, Dunod, Paris, 1966.
26. Kaufmann A., Desbazeille G., *La méthode du chemin critique*, Dunod, Paris, 1966.
27. Kompans T. S., Reiter S., *A Model of Transportation*, 1951.
28. Katz L., *An application of matrix algebre to the study of human relations*, mim. Univ. of. North Carolina, 1950.
29. Kuhn H. W., *The Hungarian Method for the assiguement problem*, Naval Research Quaterly, 1955.
30. Kai Lai Chung, *Markov chains with stationary transition probability*, Berlin, 1960.
31. Lupescu Titus, Alex. Roșu, Mihu Cerchez, *Programarea matematică*, Editura militară, București, 1965.
32. Mihoc G., Nădejde I., *Programarea matematică*, vol. I, Editura științifică, București, 1966.
33. Mihoc Gh., Firescu D., *Statistică matematică*, Editura didactică și pedagogică, București, 1966.

Scanned with OKEN Scanner

TABLA DE MATERII

CAP. I **NOŢIUNI DE BAZA ÎN TEORIA GRAFELOR**

§ 1. Introducere 9
§ 2. Matrice asociate unui graf..... 17
§ 3. Drum, circuit, lanţ, ciclu..... 21
§ 4. Graf tare conex. Graf conex. Componentă conexă a grafului. Arbori într-un graf.... 24
§ 5. Descompunerea unui graf conex şi fără circuite, în clase ordonate de vîrfuri..... 30
§ 6. Numerele de stabilitate internă şi externă. Nucleul unui graf 35
§ 7. Funcţia Grundy şi determinarea nucleelor unui graf 43

CAP. II **DRUMURI ÎNTR-UN GRAF**

§ 1. Introducere 47
§ 2. Drumul de lungime minimă între două vîrfuri ale unui graf 48
§ 3. Algoritme pentru găsirea drumului de lungime minimă dintre două vîrfuri ale unui graf 50
§ 4. Drum de lungime maximă într-un graf.... 65
§ 5. Drumuri hamiltoniene într-un graf..... 80
§ 6. Aplicaţie 101
§ 7. Arbori într-un graf. Arbore minimal..... 112

Scanned with OKEN Scanner

CAP. III MODELE DE REȚELE APLICATE ÎN ECONOMIE

A. REȚELE DE TRANSPORT

§ 1. Introducere 128
§ 2. Flux într-o rețea de transport..... 130
§ 3. Algoritme pentru determinarea fluxului maxim (minim) într-o rețea de transport.. 132
§ 4. Probleme care conduc la determinarea fluxului maxim în rețea..... 150

B. CUPLAJUL MAXIM ÎN REȚEA. PROBLEME DE AFECTARE

§ 1 Introducere 164
§ 2. Cuplajul maxim 166
§ 3. Cuplajul maxim legat de costul total al repartizărilor 174
§ 4. Algoritmul ungar pentru determinarea cuplajului maxim 183
§ 5. Aplicații ale algoritmului Egervahry-Kuhn 184
§ 6. Aplicarea algoritmului ungar pentru rezolvarea problemelor de transport..... 199

C. REȚELE PENTRU PLANIFICAREA, CONDUCEREA ȘI CONTROLUL PRODUCȚIEI

§ 1. Introducere 209
§ 2. Elementele rețelelor de activitate..... 210
§ 3. Criterii de construire a unei rețele de activitate 211
§ 4. Drumul critic 215
§ 5. Rezerve de timp în elaborarea proiectelor 220
§ 6. Găsirea drumului critic și a rezervelor de timp, legate de proiectul de construire a unui imobil 227

CAP. IV GRAFELE ȘI LEGĂTURA LOR CU ALTE DOMENII DE CERCETARE

§ 1. Grafe și lanțuri Markov..... 231
§ 2. Comportarea asimptotică a lanțului Markov (ergodicitate) 236

Scanned with OKEN Scanner

§ 3. Determinarea elementelor matricei-limită cu ajutorul calculului matriceal 240
§ 4. Corespondența dintre lanțurile Markov și grafe 245
§ 5. Condiția de ergodicitate a unui lanț Markov, prezentată cu ajutorul grafelor..... 250
§ 6. Condiția de ergodicitate în cazul lanțurilor Markov nestaționare 260
§ 7. Legătura grafelor cu teoria jocurilor..... 265

ANEXĂ

§ 1. Noțiuni de teoria mulțimilor..... 271
§ 2. Elemente de calcul matriceal..... 276
§ 3. Noțiuni de teoria probabilităților..... 281

Scanned with OKEN Scanner

Redactor : DINU-IOANIȚIU NECHITA
Tehnoredactor : MIRCEA NASTA

Dat la cules 24.II.1969. Bun de tipar 28.IV.1969. Apărut 1969. Tiraj 2350+139+30 ex. broșate. Hîrtie tipar înalt tip A 63 g/m². Format 16/54×84. Coli editoriale 12,41. Coli tipar 19.00. A 1865/1969. Indici de clasificare zecimală pentru bibliotecile mari 513.8, pentru bibliotecile mici 51

Tiparul executat la Combinatul Poligrafic „Casa Scînteii", Piața Scînteii nr. 1, București — Republica Socialistă România. Comanda nr. 90.176

Scanned with OKEN Scanner